ISSN:0975-9646

Optimized Mobile Search Engine

E.Chaitanya[1], Dr.Sai Satyanarayana Reddy[2], O.Srinivasa Reddy[3]

[1]*M.Tech, CSE, LBRCE, Mylavaram,*
[2]*Professor, CSE, LBRCE, Mylavaram, India,*
[3]*Asst.professor,CSE,LBRCE,Mylavaram,India.*

Abstract-We propose a Optimized Personal Search Engine for mobile , that takes the users' preferences and data in the form of concepts by analyzing their click through data. I give the importance of location information in mobile search, so my Optimized Personal Search Engine classifies these concepts like content concepts and location concepts. In addition, users locations (positioned by GPS) are used to give the location concepts in Optimized Personal Search Engine . The user preferences are organized in an ontology-based and the multi-facet user profile is used to adapt a personalized ranking function for rank adaptation of future search results. To characterize the concepts associated with a query and their relevance's to the users need and four entropies are introduced to balance the weights between the content and location facets. Based on the client-server model, i also present a detailed architecture and design for implementation of Optimized Personal Search Engine. In our design, the client collects and stores locally the click through data and to protect privacy such as heavy tasks such as concept extraction, training and reran king are performed at the Optimized Personal Search Engine server. Moreover, I address the privacy issue by restricting the information in the user profile exposed to the Optimized Personal Search Engine server with two privacy parameters. We prototype Optimized Personal Search Engine on the Google Android platform. Experimental results show that Optimized Personal Search Engine significantly improves the precision comparing to the baseline.

Keyterms:- Optimized mobile Search Engine, mobile, Client-Server Model, location and content based

1.INTRODUCTION ABOUT MOBILE SEARCH:-

The major and main problem in mobile search is that the interactions between the users and search engines are limited by the small form factors of the mobile devices. For that result, mobile users tend to submit shorter, and hence, there are more ambiguous queries compared to their web search counter parts. So in order to return highly relevant results to the users, the mobile search engines must be able to profile the users' interests and personalize the search results according to the users' profiles. One of the practical approach to capturing a user's interests for personalization is to analyze the user's Click through data. The researcher Leung developed a search engine personalization method based on users' concept preferences and showed that it is more effective and most of the previous work assumed that all concepts are of the same type. we Observing the need for different types of concepts and we present in this paper a *Optimized Personal Search Engine,* that represents different types of concepts in different ontology's. we recognizing the importance of location information in mobile search, we are separate concepts into location concepts and content concepts. A major problem in mobile search is that the interactions between the users and search engines are limited by the small form factors of the mobile devices. So as a result, the mobile users tend to submit shorter, and hence, there are more ambiguous queries compared to their web search counterparts. So In order to return highly relevant results to the users, the mobile search engines must be able to profile the users' interests and personalize the search results according to the users' profiles.

The practical approach to capturing a user's interests for personalization is to analyze the user's click through data. The Leung et al. was developed a search engine personalization method based on users' concept preferences and showed that it is more effective than methods that are based on page preferences. So most of the previous work assumed that all concepts are of the same type and Observing the need for different types of concepts, So we present in this paper a *Optimized Personal Search Engine* which represents different types of concepts in different ontology's. In particular, we recognizing the importance of location information in mobile search, we just separate concepts into location concepts and content concepts like that. For example, the user who is planning to visit Japan may issue the query "hotel," and then click on the search results about hotels in Japan. So From the click through of the query "hotel," *Optimized Personal Search Engine* can learn the user's content preference ("room rate" and "facilities") and location preferences ("Japan"). Accordingly, the *Optimized Personal Search Engine* will favor results that are concerned with hotel information in Japan for future queries on "hotel." The introduction of location preferences offers *Optimized Personal Search Engine* an additional dimension for capturing a user's interest and an opportunity to enhance search quality for users.

So to incorporate context information revealed by user mobility, and we also take into account the visited physical locations of users in the *Optimized Personal Search Engine*. And since this information can be conveniently obtained by GPS devices, and it is hence referred to as GPS locations. The GPS locations play an important role in mobile web search. For example, let if the user, and who is searching for hotel information, that is currently located in "Shinjuku, Japan," and his/her position can be used to personalize the search results to favor information about nearby hotels. There, we can see that the GPS locations (i.e., "Shinjuku, Tokyo") help reinforcing the user's location preferences (i.e., "Japan") derived from a user's search activities to provide the most relevant results. And our proposed framework is capable of combining a user's

GPS locations and location preferences into the personalization process. So to the best of my knowledge, our paper is the first to propose a personalization framework that utilizes a user's content preferences and location preferences as well as the GPS locations in personalizing search results.

In this paper, we propose a realistic design for *Optimized Personal Search Engine* by adopting the meta search approach which replies on one of the commercial search engines like Google, Yahoo and Bing to perform an actual search. So the client is responsible for receivingthe user's requests, submitting the requests to the *Optimized Personal Search Engine* server, displaying the returned results and collecting his/her click throughs in order to derive his/her personal preferences. The *Optimized Personal Search Engine* server, on the other hand, is responsible for handling heavy tasks such as forwarding the requests to a commercial search engine as well as training and reranking of search results before they are returned to the client. So The user profiles for specific users are stored on the *Optimized Personal Search Engine* clients, thus preserving privacy to the users. *Optimized Personal Search Engine* has been prototyped with *Optimized Personal Search Engine* clients on the Google Android platform and the *Optimized Personal Search Engine* server on a PC server to validate the proposed ideas.

The main research of this paper are as follows:

1. It studies the unique characteristics of content and location concepts, and that provides a coherent strategy using a client-server architecture to integrate them into a uniform solution for the mobile environment.
2. The proposed *Optimized Personal Search Engine* is an innovative approach for personalizing web search results. So by mining content and location concepts for user profiling, and it utilizes both the content and location preferences to personalize search results for a user.
3. *The Optimized Personal Search Engine* incorporates a user's physical locations in the personalization process. So we conduct experiments to study the influence of a user's GPS locations in personalization. So the results show that GPS locations helps improve retrieval effectiveness for location queries (i.e., queries that retrieve lots of location information).
4. We propose a new and realistic system design for *Optimized Personal Search Engine*. Our design adopts the server-client model in which user queries are forwarded to a *Optimized Personal Search Engine* server for processing the training and reranking quickly. So we implement a working prototype of the *Optimized Personal Search Engine* clients on the Google Android platform, and the *Optimized Personal Search Engine* INE server on a PC to validate the proposed ideas. The empirical results show that our design can efficiently handle user requests.
5. Privacy preservation is a challenging issue in *Optimized Personal Search Engine* where users send their user profiles along with queries to the *Optimized Personal Search Engine* server to obtain personalized search results. *Optimized Personal Search Engine* addresses the privacy issue by allowing users to control their privacy levels with two privacy parameters, minimum Distance and expRatio. The Empirical results show that our proposal facilitates smooth privacy preserving control while maintaining good ranking quality.
6. We conduct a comprehensive set of experiments to evaluate the performance of the proposed *Optimized Personal Search Engine*. Empirical results show that the ontology-based user profiles can successfully capture users' content and location preferences and utilize the preferences to produce relevant results for the users. So it significantly outperforms existing strategies which use either content or location preference only.

2 .RELATED WORK

Click through data have been used in determining the users' preferences on their search results. click through data for the query "hotel," composes of the search results and the ones that the user clicked on . As shown, cis are the content concepts and lis are the location concepts extracted from the corresponding results. So Many existing personalized web search systems are based click through data to determine users' preferences and Joachims proposed to mine document preferences from click through data. Later, Ng et al. proposed to combine a spying technique together with a novel voting procedure to determine user preferences. And more recently Leung et al introduced an effective approach to predict users' conceptual preferences from click through data for personalized query suggestions. So Search queries can be classified as content (i.e., non-geo) or location (i.e., geo) queries. It was found that a significant number of queries were location queries focusing on location information. So In order to handle the queries that focus on location information, the number of location-based search systems designed for location queries have been proposed. Yokoji proposed a location-based search system for web documents. The Location information was extracted from the web documents, those which was converted into latitude-longitude pairs . So when a user submits a query together with a latitude-longitude pair, and the system creates a search circle centered at the specified latitude-longitude pair and retrieves documents containing location information within the search circle. Later on, Chen et al. studied the problem of efficient query processing in location-based search systems. The query is assigned with a query footprint that specifies the geographical area of interest to the user. There are Several algorithms are employed to rank the search results as a combination of a textual and a geographic score. Li et al proposed a probabilistic topic-based framework for location sensitive domain information retrieval. So instead of modeling locations in latitude-longitude pairs, this model assumes that users can be interested in a set of location sensitive topics. So It recognizes the geographical influence distributions of topics and models it using probabilistic Gaussian Process classifiers.

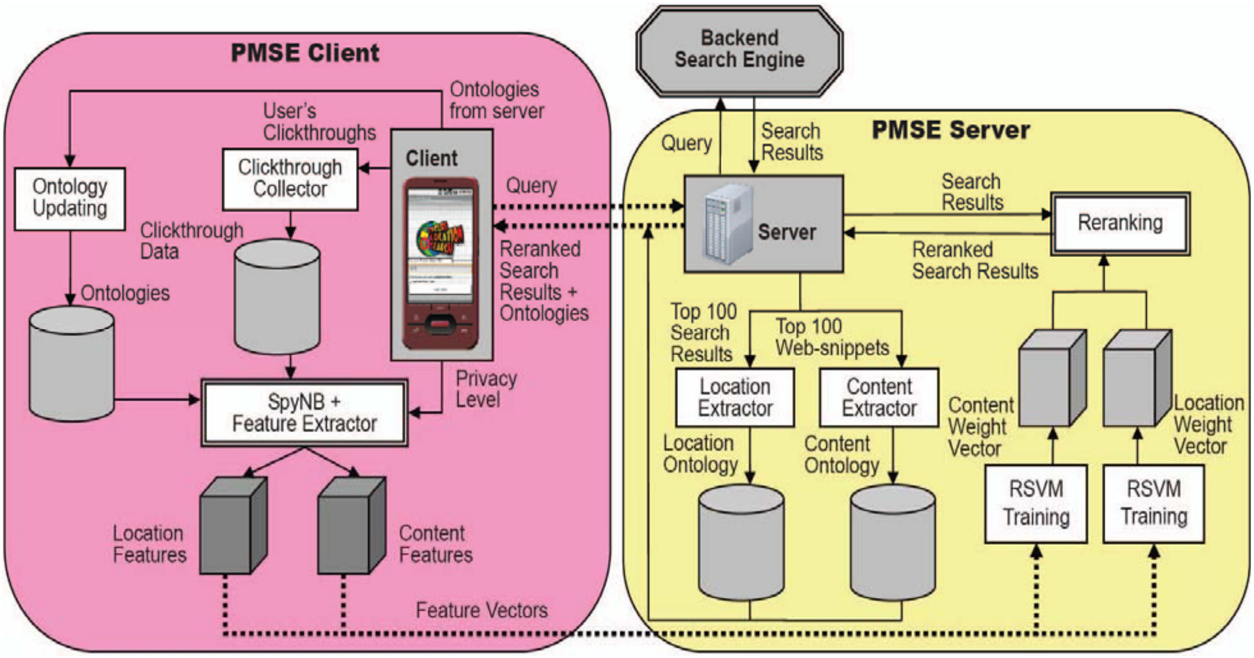

Fig. 1. The general process flow of Optimized Personal Search Engine

The differences among existing works and ours is

1. Most existing location-based search systems, such as , require users to manually define their location preferences (with latitude-longitude pairs or text form) or to manually prepare a set of location sensitive topics. *Optimized Personal Search Engine* profiles both of the user's content and location preferences in the ontology based user profiles which are automatically learned from the click through and GPS data without requiring extra efforts from the user.
2. We propose and implement a new and realistic design for *Optimized Personal Search Engine*. To train the user profiles quickly and efficiently our design forwards user requests to the *Optimized Personal Search Engine* server to handle the training and re ranking processes.
3. Existing works on personalization do not address the issues of privacy preservation. *Optimized Personal Search Engine* addresses this issue by controlling the amount of information in the client's user profile being exposed to the *Optimized Personal Search Engine* server using two privacy parameters which can control privacy smoothly while maintaining good ranking quality.

3. System Design

Optimized Personal Search Engine's client-server architecture, that which meets three important requirements. And the First, computation-intensive tasks such as RSVM training, should be handled by the *Optimized Personal Search Engine Engine* server due to the limited computational power on mobile devices. The Second, data transmission between client and server should be minimized to ensure fast and efficient processing of the search. The Third, click through data representing precise user preferences on the search results, should be stored on the *Optimized Personal Search Engine Engine* clients in order to preserve user privacy. In the *Optimized Personal Search Engine Engine*'s client-server architecture, *Optimized Personal Search Engine Engine* clients are responsible for storing the user click throughs and the ontologies derived from the *Optimized Personal Search Engine Engine* server. Simple tasks, such as updating click thoughs and ontologies, that creating feature vectors and displaying re ranked search results are handled by the *Optimized Personal Search Engine Engine* clients with limited computational power and On the other hand heavy tasks such as RSVM training and reranking of search results, are handled by the *Optimized Personal Search Engine Engine* server. Moreover, in order to minimize the data transmission between client and server, the *Optimized Personal Search Engine Engine* client would only need to submit a query together with the feature vectors to the *Optimized Personal Search Engine Engine* server, and the server would automatically return a set of reranked search results according to the preferences stated in the feature vectors. And the data transmission cost is minimized, because only the essential data (i.e., query, feature vectors, ontologies and search results) are transmitted between client and server during the personalization process. *Optimized Personal Search Engine Engine*'s design addressed the issues:

1) limited computational power on mobile devices and

2) data transmission minimization.

Optimized Personal Search Engine consists of two major activities:

1. The Re ranking the search results at *Optimized Personal Search Engine* server. When a user submits a query on the *Optimized Personal Search Engine* client and the query together with the feature vectors containing the user's content and location preferences (that is filtered ontologies according to the user's privacy setting) are forwarded to the

Optimized Personal Search Engine server and which in turn obtains the search results from the back-end search engine (eg. Google). The content and location concepts are extracted from the search results and organized into ontologies to capture the relationships between the concepts. So the server is used to perform ontology extraction for its speed. These feature vectors from the client are then used in RSVM training to obtain a content weight vector and a location weight vector that representing the user interests based on the user's content and location preferences for the re ranking and Again, the training process is performed on the server for its speed. So the search results are then re ranked according to the weight vectors obtained from the RSVM training. So Finally, the re ranked results and the extracted ontologies for the personalization of future queries are returned to the client.

2. Ontology update and click through collection at *Optimized Personal Search Engine* client. The ontologies returned from the *Optimized Personal Search Engine* server contain the concept space that models the relationships between the concepts extracted from the search results and they are stored in the ontology database on the client.

1 When the user clicks on a search result the click through data together with the associated content and location concepts are stored in the click through database on the client. The click throughs are stored on the *Optimized Personal Search Engine* clients. So the *Optimized Personal Search Engine* server does not know the exact set of documents that the user has clicked on. And this design allows user privacy to be preserved in certain degree. Two privacy parameters minDistance and expRatio, are proposed to control the amount of personal preferences exposed to the *Optimized Personal Search Engine* server. If the user is concerned with his/her own privacy and the privacy level can be set to high so that only limited personal information will be included in the feature vectors and passed along to the *Optimized Personal Search Engine* server for the personalization. On the other hand if the user wants more accurate results according to his/her preferences and the privacy level can be set to low so that the *Optimized Personal Search Engine* server can use the full feature vectors to maximize the personalization effect.

Since the ontologies can be derived online at the *Optimized Personal Search Engine* server and an alternative system design is for the user to pass only the click through data to the *Optimized Personal Search Engine* server and that is to perform both feature extraction and RSVM training on the *Optimized Personal Search Engine* server to train the weight vectors for reranking. And however, if all those click throughs are exposed to the *Optimized Personal Search Engine* server, and the server would know exactly what the user has clicked. To address privacy issues, click throughs are stored on the *Optimized Personal Search Engine* client, and the user could adjust the privacy parameters to control the amount of personal information to be included in the feature vectors, which are forwarded to the *Optimized Personal Search Engine* server for RSVM training to adapt personalized ranking functions for content and location preferences.

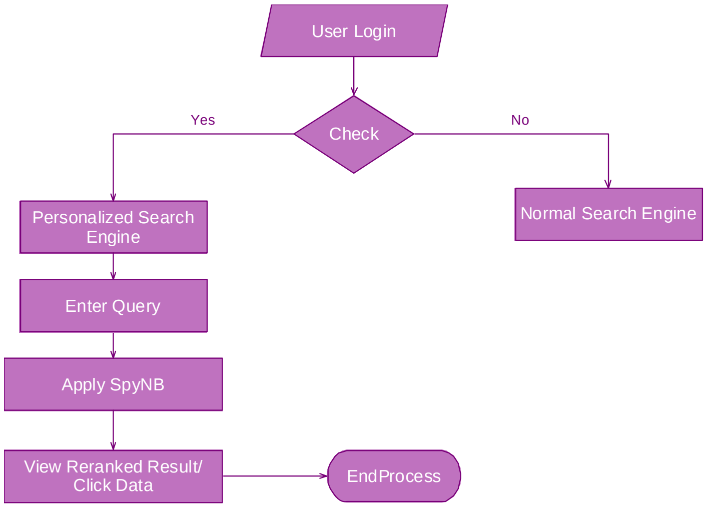

Fig. 2. The data flow diagram of Optimized Personal Search Engine

4.USER INTEREST PROFILING

Optimized Personal Search Engine uses "concepts" to model the interests and preferences of a user. And the location information is important in mobile search the concepts are further classified into two different types namely content concepts and location concepts. Further these concepts are modeled as ontologies in order to capture the relationships between the concepts. And then we observe that the characteristics of the *content concepts* and *location concepts* are different. So we propose two different techniques for building the content ontology and location ontology. Here the ontologies indicate a possible concept space arising from a user's queries, which are maintained along with the click through data for future preference adaptation. So in *Optimized Personal Search Engine*, we have adopt ontologies to model the concept space because they not only can represent concepts but also capture the relationships between concepts. Due to different characteristics of the content concepts and location concepts

4.1 The content Ontology

Our content concept extraction method first extracts all the keywords and phrases (excluding the stop words) from the web-snippets to arising from q. If a keyword/phrase exists frequently in the web snippets arising from the query q we would treat it as an important concept related to the query as it coexists in close proximity with the query in the top documents. So these following shows that the support formulae which is inspired by the well known problem of finding frequent item sets in data mining is employed to measure the importance of a particular keyword/phrase ci with respect to the query q:

$$support(c_i) = \frac{sf(c_i)}{n} \cdot |c_i|,$$

where sf(ci) is the snippet frequency of the keyword/ phrase ci (that is the number of web-snippets containing ci) and n is the number of web-snippets returned and |ci| is the number of terms in the keyword/phrase ci. If the support of a keyword/phrase ci is higher than the threshold s (s ¼ 0:03

in our experiments), we treat ci as a concept for q. We adopt the following two propositions to determine the relationships between concepts for ontology formulation:

1.**Similarity**:- Two concepts which coexist a lot on the search results might represent the same topical interest.

2. **Parent-child relationship**:- More specific concepts often appear with general terms, while the reverse is not true. we mark ci as cj's child. For example, the more specific concept "meeting facility" tends to occur together with the general concept "facilities," while the general concept "facilities" might also occur with concepts such as "meeting room" or "swimming pool," i.e., not only with the concept "meeting facility."

The given example content ontology created for the query "hotel," where content concepts linked with a one sided arrow (!) are parent-child concepts, and concepts linked with a double-sided arrow ($) are similar concepts. Fig. 2 shows the possible concept space determined for the query "hotel," while the click through data determine the user preferences on the concept space. In general, the ontology covers more than what the user actually wants. The concept space for the query "hotel" consists of "map," "reservation," "room rate,"..., etc. If the user is indeed interested in information about hotel rates and clicks on pages containing "room rate" and "special discount rate" concepts, the captured click through favors the two clicked concepts. Feature vectors containing the concepts "room rate" and "special discount rate" as positive preferences will be created corresponding to the query "hotel." As indicated in Fig. 2, when the query is issued again later, these feature vectors will be transmitted to the *Optimized Personal Search Engine* server and transformed into a content weight vector to rank the search results according to the user's content preferences.

Table 1. Statistics of the Location Ontology

No. of Countries	7	Total No. of Nodes	16899
No. of Regions	190	Country-Region Edges	190
No. of Provinces	6699	Region-Province Edges	1959
No. of Towns	10003	Province-City Edges	14897

4.2 Location Ontology

Our approach for extracting location concepts is different from that for extracting content concepts. We observe two important issues in location ontology formulation. First, the document usually embodies only a few location concepts and thus only very few of them co-occur with the query terms in web-snippets. So to alleviate this problem, we can extract location concepts from the full documents. And second, similarity and parent-child relationship cannot be accurately derived statistically because the limited number of location concepts embodied in documents. So Furthermore many geographical relationships among locations have already been captured as facts. Thus, we can obtain about 17,000 city, region, and then country names and, create predefined location ontology among these locations. So we organize all the cities as children under their provinces and all the provinces as children under their regions and all the regions as children under their countries. The predefined location ontology is used to associate location information with the searching results. All of the keywords and key-phrases from the documents returned for query q and are extracted. If a keyword or key-phrase in a retrieved document d matches a location name in our pre defined location ontology, then it will be treated as a location concept of d. For example, we assume that document d contains the keyword "Los Angeles." And it would then be matched against the location ontology. Since "Los Angeles" is a location in our location ontology, it is treated as a location concept related to d. Furthermore, we would explore the predefined location hierarchy, which would identify "Los Angeles" as a city under the state "California." Thus, the location "/United States/California/Los Angeles/" is associated with document d. If a concept matches several nodes in the location ontology and all matched locations will be associated with the document.

Similar to the content ontology the location ontology together with click through data are used to create feature vectors containing the user location preferences. They will then be transformed into a location weight vector to rank the search results according to the user's location references.

5. PERSONALIZED RANKING FUNCTIONS

Upon reception of the user's preferences the ranking SVM is employed to learn a personalized ranking function for rank adaptation of the search results according to the user content and location preferences. For a given query, the set of content concepts and a set of location concepts are extracted from the search results as the document features. Since each document can be represented by a feature vector and it can be treated as a point in the feature space. Using this preference pairs as the input and RSVM aims at finding a linear ranking function and which holds for as many document preference pairs as possible. An adaptive implementation SVM light available and is used in our experiments. In the following, we are discussing two issues in the RSVM training process:

1) how to extract the feature vectors for a document

2) how to combine the content and location weight vectors into one integrated weight vector.

5.1 Extracting Features for Training

We propose two feature vectors namely, the content feature vector and location feature vector to represent the content and location information associated with documents. The feature vectors are extracted by taking into account the concepts existing in documents and other related concepts in the ontology of the query. For example let if a document dk embodies the content concept ci and location concept li and the weight of component ci in the content feature vector of document dk is incremented by one and the weight of li in the location feature vector is incremented by one. The similarity and parent-child relationships of the concepts in the extracted

Concept of ontologies are also incorporated in the training based on the following four different types of relationships:

1. Similarity
2. Ancestor
3. Descendant and
4. Sibling

1. Content feature vector: If content concepts ci is in a web-snippet sk and their values are incremented in the content feature vector with the following equation:

$$\forall c_i \in s_k, \phi_C(q, d_k)[c_i] = \phi_C(q, d_k)[c_i] + 1.$$

For other content concepts cj that are related to the content concept ci (either they are similar or cj is the ancestor/descendant/sibling of ci) in the content ontology and they are incremented in the content feature vector according to the following equation:

$$\begin{aligned}\forall c_i \in s_k, \phi_C(q, d_k)[c_j] &= \phi_C(q, d_k)[c_j] \\ &+ sim_R(c_i, c_j) + ancestor(c_i, c_j) \\ &+ descendant(c_i, c_j) + sibling(c_i, c_j).\end{aligned}$$

2. Location feature vector: If location concept li is in a web-snippet dk and its value is incremented in the location feature vector with the following equation:

$$\forall l_i \in d_k, \phi_L(q, d_k)[l_i] = \phi_L(q, d_k)[l_i] + 1.$$

For other location concepts lj that are related to the concept li(lj is the ancestor/descendant
li) in the location ontology and they are incremented in the location feature vector according to the following equation:

$$\begin{aligned}\forall l_i \in d_i,\ \phi_L(q, d_k)[l_j] &= \phi_L(q, d_k)[c_j] + ancestor(l_i, l_j) \\ &+ descendant(l_i, l_j) + sibling(l_i, l_j).\end{aligned}$$

5.2 GPS Data and Combination of Weight Vectors

The content feature vector together with the document preferences obtained from SpyNB are served as input to RSVM training to obtain the content weight vector. The location weight vector is obtained similarly using the location feature vector and the document preferences and represent the content and location user profiles for a user u on a query q in our method.

GPS locations are important information that can be useful in personalizing the search results. For example the user may use his/her mobile device to find movies on show in the nearby cinemas. Thus the PMSE incorporates the GPS locations into the personalization process by tracking the visited locations. This function is realized by the embedded GPS modules on the PMSE client. We believe that users are possibly interested in locations where they have visited. Thus our goal is to integrate the factor of GPS locations in to reflect the possible preferences. So, if a user has visited the GPS location the weight of the location concept in is incremented according the following equation:

$$\forall l_r \text{ that } u \text{ has visited}, \overrightarrow{w_{L,q,u}}[l_r] = \overrightarrow{w_{L,q,u}}[l_r] + w_{GPS}(u, l_r, t_r),$$

The weight being added to the according to the following decay equation

$$w_{GPS}(l_r, t_r) = w_{GPS\_0} \cdot e^{-t_r},$$

The set of location concepts flsg that are closely related to the GPS location lr (ls is the ancestor/descendant/sibling of lr) in the location ontology are also possible candidates that the user may be interested in. So, the weight of the location concept ls in the weight vector incremented according to the following equation

$$\begin{aligned}\forall l_r\ \overrightarrow{w_{L,q,u}}[l_s] &= \overrightarrow{w_{L,q,u}}[l_s] + w_{GPS}(u, l_r, t_r) \\ &\cdot (ancestor(l_i, l_j) + descendant(l_i, l_j) + sibling(l_i, l_j)).\end{aligned}$$

Optimized Personal Search Engine will rank the documents in the returned search according to the following above formulae. So we can get the search results easily by using this *Optimized Personal Search Engine*

6. Conclusions

We proposed *Optimized Personal Search Engine* to extract and learn a user's content and location preferences based on the user's click through. So to adapt to the user mobility, and we incorporated the user's GPS locations in the personalization process. We observed that GPS locations help to improve retrieval effectiveness, and especially for location queries. We also proposed two privacy parameters are minDistance and expRatio, to address privacy issues in *Optimized Personal Search Engine* by allowing users to control the amount of personal information exposed to the *Optimized Personal Search Engine* server. The privacy parameters facilitate smooth control of privacy exposure while maintaining good ranking quality. For future work we will investigate methods to exploit regular travel patterns and query patterns from the GPS and clickthrough data to further enhance the personalization effectiveness of *Optimized Personal Search Engine*.

References

[1] Appendix, http://www.cse.ust.hk/faculty/dlee/tkde-pmse/appendix.pdf, 2012.

[2] Nat'l geospatial, http://earth-info.nga.mil/, 2012.

[3] svmlight, http://svmlight.joachims.org/, 2012.

[4] World gazetteer, http://www.world-gazetteer.com/, 2012.

[5] E. Agichtein, E. Brill, and S. Dumais, "Improving Web Search Ranking by Incorporating User Behavior Information," Proc. 29th Ann. Int'l ACM SIGIR Conf. Research and Development in Information Retrieval (SIGIR), 2006.

[6] E. Agichtein, E. Brill, S. Dumais, and R. Ragno, "Learning User Interaction Models for Predicting Web Search Result Preferences," Proc. Ann. Int'l ACM SIGIR Conf. Research and Development in Information Retrieval (SIGIR), 2006.

[7] Y.-Y. Chen, T. Suel, and A. Markowetz, "Efficient QueryProcessing in Geographic Web Search Engines," Proc. Int'l ACM SIGIR Conf. Research and Development in Information Retrieval(SIGIR), 2006.

[8] K.W. Church, W. Gale, P. Hanks, and D. Hindle, "Using Statistics in Lexical Analysis," Lexical Acquisition: Exploiting On-Line Resources to Build a Lexicon, Psychology Press, 1991

[9] Q. Gan, J. Attenberg, A. Markowetz, and T. Suel, "Analysis of Geographic Queries in a Search Engine Log," Proc. First Int'l Workshop Location and the Web (LocWeb), 2008.

[10] T. Joachims, "Optimizing Search Engines Using Clickthrough Data," Proc. ACM SIGKDD Int'l Conf. Knowledge Discovery and Data Mining, 2002.

[11] K.W.-T. Leung, D.L. Lee, and W.-C. Lee, "Personalized Web Search with Location Preferences," Proc. IEEE Int'l Conf. Data Mining (ICDE), 2010.

[12] K.W.-T. Leung, W. Ng, and D.L. Lee, "Personalized Concept-Based Clustering of Search Engine Queries," IEEE Trans. Knowledge and Data Eng., vol. 20, no. 11, pp. 1505-1518, Nov. 2008.

[13] H. Li, Z. Li, W.-C. Lee, and D.L. Lee, "A Probabilistic Topic-Based Ranking Framework for Location-Sensitive Domain Information Retrieval," Proc. Int'l ACM SIGIR Conf. Research and Development in Information Retrieval (SIGIR), 2009.

[14] B. Liu, W.S. Lee, P.S. Yu, and X. Li, “Partially Supervised Classification of Text Documents,” Proc. Int’l Conf. Machine Learning (ICML), 2002.
[15] W. Ng, L. Deng, and D.L. Lee, “Mining User Preference Using Spy Voting for Search Engine Personalization,” ACM Trans. Internet Technology, vol. 7, no. 4, article 19, 2007.
[16] J.Y.-H. Pong, R.C.-W. Kwok, R.Y.-K. Lau, J.-X. Hao, and P.C.-C. Wong, “A Comparative Study of Two Automatic Document Classification Methods in a Library Setting,” J. Information Science, vol. 34, no. 2, pp. 213-230, 2008.
[17] C.E. Shannon, “Prediction and Entropy of Printed English,” Bell Systems Technical J., vol. 30, pp. 50-64, 1951.
[18] Q. Tan, X. Chai, W. Ng, and D. Lee, “Applying Co-Training to Clickthrough Data for Search Engine Adaptation,” Proc. Int’l Conf. Database Systems for Advanced Applications (DASFAA), 2004.
[19] J. Teevan, M.R. Morris, and S. Bush, “Discovering and Using Groups to Improve Personalized Search,” Proc. ACM Int’l Conf. Web Search and Data Mining (WSDM), 2009. [20] E. Voorhees and D. Harman, TREC Experiment and Evaluation in Information Retrieval. MIT Press, 2005.
[21] Y. Xu, K. Wang, B. Zhang, and Z. Chen, “Privacy-Enhancing Personalized Web Search,” Proc. Int’l Conf. World Wide Web (WWW), 2007.
[22] . Yokoji, “Kokono Search: A Location Based Search Engine,” Proc.Int’l Conf. World Wide Web (WWW), 200