Android Based Effective and Efficient Search Engine Retrieval System Using Ontology

S.Karthika[1], S.Gunanandhini[2], Mr.A.Vijayanarayanan[3]

CSE, Srinivasa Institute of Engineering and Technology

Abstract

A personalized mobile search engine, PMSE, that captures the users' preferences in the form of concepts by mining their clickthrough data. Due to the importance of location information in mobile search, PMSE classifies these concepts into content concepts and location concepts. In addition, users' locations (positioned by GPS) are used to supplement the location concepts in PMSE. The user preferences are organized in an ontology-based, multi-facet user profile, which are used to adapt a personalized ranking function for rank adaptation of future search results. Based on the client-server model, we also present a detailed architecture and design for implementation of PMSE. In our design, the client collects and stores locally the clickthrough data to protect privacy, whereas heavy tasks such as concept extraction, training and re-ranking are performed at the PMSE server.

1. Introduction

A major problem in mobile search is that the interactions between users and search engines are limited by the small form of the mobile devices. As a result, mobile users tend to submit shorter, hence, more ambiguous queries compared to their web search counterparts. In order to return highly relevant results to the users, mobile search engines must be able to profile the users' interests and personalize the search results according to the users' profiles.

A practical approach to capturing a user's interests for personalization is to analyze the user's clickthrough data Leung, et. al., 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. However, most of the previous work assumed that all concepts are of the same type. Observing the need for different types of concepts, we present in this paper a personalized mobile search engine, PMSE, which represents different types of concepts in different ontologies. In particular, recognizing the importance of location information in mobile search, we separate concepts into location concepts and content concepts. For example, a user who is planning to visit Japan may issue the query "hotel", and click on the search results about hotels in Japan. From the clickthroughs of the query "hotel", PMSE can learn the user's content preference (e.g., "room rate" and "facilities") and location preferences ("Japan"). Accordingly, PMSE will favor results that are concerned with hotel information in Japan for future queries on "hotel". The introduction of location preferences offers PMSE an additional dimension for capturing a user's interest and an opportunity to enhance search quality for users.

In this paper, we propose a realistic design for PMSE by adopting the metasearch approach which replies on one of the commercial search engines, such as Google, Yahoo or Bing, to perform an actual search. The client is responsible for receiving the user's requests, submitting the requests to the PMSE server, displaying the returned results, and collecting his/her clickthroughs in order to derive his/her personal preferences. The PMSE 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. The user profiles for specific users are stored on the PMSE clients, thus preserving privacy to the users. PMSE has been prototyped with PMSE clients on the Google Android platform and the PMSE server on a PC server to validate the proposed ideas.

We also recognize that the same content or location concept may have different degrees of importance to different users and different queries. To formally characterize the diversity of the concepts associated with a query and their relevances to the user's need, we introduce the notion of content and location entropies to measure the amount of content and location information associated with a query. Similarly, to measure how much the user is interested in the content and/or location information in the results, we propose click content and location entropies. Based on these entropies, we develop a method to estimate the personalization effectiveness for a particular query of a given user, which is then used to strike a balanced combination between the content and location preferences. The results are reranked according to the user's content and location preferences before returning to the client.

The main contributions of this paper are as follows:

1.This paper studies the unique characteristics of content and location concepts, and provides a coherent strategy using a client-server architecture to integrate them into a uniform solution for the mobile environment.

2.The proposed personalized mobile search engine, PMSE, is an innovative approach for personalizing web search results. By mining content and location concepts for user profiling, it utilizes both the content and location preferences to personalize search results for a user.

3.PMSE incorporates a user's physical locations in the personalization process. We conduct experiments to study the influence of a user's GPS locations in personalization. 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 PMSE. Our design adopts the server-client model in which user queries are forwarded to a PMSE server for processing the training and reranking quickly. We implement a working prototype of the PMSE clients on the Google Android platform, and the PMSE server on a PC to validate the proposed ideas. Empirical results show that our design can efficiently handle user requests.

5.Privacy preservation is a challenging issue in PMSE, where users send their user profiles along with queries to the PMSE server to obtain personalized search results. PMSE addresses the privacy issue by allowing users to control their privacy levels with two privacy parameters, minDistance and expRatio. 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 PMSE. 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. It significantly out-performs existing strategies which use either content or location preference only.

2. System Design

Figure 1 shows PMSE's client-server architecture, which meets three important requirements. First, computation intensive tasks, such as RSVM training, should be handled by the PMSE server due to the limited computational power on mobile devices. Second, data transmission between client and server should be minimized to ensure fast and efficient processing of the search. Third, clickthrough data, representing precise user preferences on the search results, should be stored on the PMSE clients in order to preserve user privacy.

In the PMSE's client-server architecture, PMSE clients are responsible for storing the user clickthroughs and the ontologies derived from the PMSE server. Simple tasks, such as updating clickthoughs and ontologies, creating feature vectors, and displaying reranked search results are handled by the PMSE clients with limited computational power. On the other hand, heavy tasks, such as RSVM training and reranking of search results, are handled by the PMSE server. Moreover, in order to minimize the data transmission between client and server, the PMSE client would only need to submit a query together with the feature vectors to the PMSE server, and the server would automatically return a set of reranked search results according to the preferences stated in the feature vectors. 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. PMSE's design addressed the issues: (1) limited computational power on mobile devices, and (2) data transmission minimization.

PMSE consists of two major activities: 1) Reranking the search results at the PMSE server, and 2) Ontology update and clickthrough collection at a mobile client.

1) Reranking the search results at PMSE server: When a user submits a query on the PMSE client, the query together with the feature vectors containing the user's content and location preferences (i.e., filtered ontologies according to the user's privacy setting) are forwarded to the PMSE server, which in turn obtains the search results from the backend search engine

(i.e., Google). The content and location concepts are extracted from the search results and organized into ontologies to capture the relationships between the concepts. The server is used to perform ontology extraction for its speed. The feature vectors from the client are then used in RSVM training to obtain a content weight vector and a location weight vector, representing the user interests based on the user's content and location preferences for the reranking. Again, the training process is performed on the server for its speed. The search results are then reranked according to the weight vectors obtained from the RSVM training. Finally, the reranked results and the extracted ontologies for the personalization of future queries are returned to the client.

2) Ontology update and clickthrough collection at PMSE client: The ontologies returned from the PMSE server contain the concept space that models the relationships between the concepts extracted from the search results. They are stored in the ontology database on the client1 . When the user clicks on a search result, the clickthrough data together with the associated content and location concepts are stored in the clickthrough database on the client. The clickthroughs are stored on the PMSE clients, so the PMSE server does not know the exact set of documents that the user has clicked on. 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 PMSE server. If the user is concerned with his/her own privacy, 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 PMSE server for the personalization. On the other hand, if a user wants more accurate results according to his/her preferences, the privacy level can be set to low so that the PMSE server can use the full feature vectors to maximize the personalization effect.

Since the ontologies can be derived online at the PMSE server, an alternative system design is for the user to pass only the clickthrough data to the PMSE server, and to perform both feature extraction and RSVM training on the PMSE server to train the weight vectors for reranking. However, if all clickthroughs are exposed to the PMSE server, the server would know exactly what the user has clicked. To address privacy issues, clickthroughs are stored on the PMSE 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 PMSE server for RSVM training to adapt personalized ranking functions for content and location preferences.

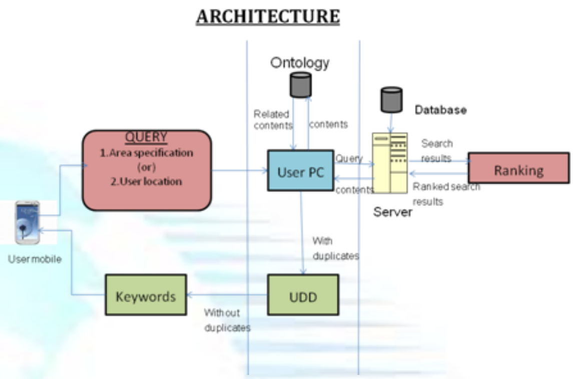

Fig. 1. The general process flow of PMSE

3.User Interest Profiling

PMSE uses "concepts" to model the interests and preferences of a user. Since location information is important in mobile search, the concepts are further classified into two different types, namely, content concepts and location concepts. The concepts are modeled as ontologies, in order to capture the relationships between the concepts. We observe that the characteristics of the content concepts and location concepts are different. Thus, we propose two different techniques for building the content ontology and location ontology. The ontologies indicate a possible concept space arising from a user's queries, which are maintained along with the clickthrough data for future preference adaptation. In PMSE, we adopt ontologies to model the concept space because they not only can represent concepts but also capture the relationships between concepts. Due to the different characteristics of the content concepts and location concepts the results can be searched efficiently for the users.

4. User Preferences Extraction and Privacy Preservation

Given that the concepts and clickthrough data are collected from past search activities, user's preference can be learned. These search preferences, inform of a set of feature vectors, are to be submitted along with future queries to the PMSE server for search result re-ranking. Instead of transmitting all the detailed personal preference information to the server, PMSE allows the users to control the amount of personal information exposed. In this section, we first review a preference mining algorithms, namely SpyNB Method, that we adopt in PMSE, and then discuss how PMSE preserves user privacy.

User preference pair can be obtained as follows,

$$d_i < d_j, \forall l_i \in P, l_j \in PN \qquad (1)$$

The preference pairs together with the extracted ontologies are used to derive a set of feature vectors on the PMSE client for submission along with future queries to the PMSE server which in turn finds a linear ranking function that best describes the user preferences using RSVM. In our client-server model, the click histories are entirely stored on the PMSE clients as shown in Figure 1. The backend search engine has no knowledge of a user's click history. Hence, the user's privacy is ensured. The PMSE server is a trusted server, which would not store all the clickthrough data. It is aware of the user's preferences, but the how much it knows is controlled by the privacy settings set by the client. The PMSE client stores the user's clickthrough and has control on the privacy setting. It would create a feature vector based on its clickthrough data and the filtered ontology according to the privacy settings at different expRatio. The feature vector is then forwarded to the PMSE server for the personalization. Thus, the PMSE server only knows about the filtered concepts that the client prefers in the form of a feature vector.

To control the amount of personal information exposed out of users' mobile devices, PMSE filters the ontologies according to the user's privacy level setting, which are specified with two privacy parameters, minDistance and expRatio. For example, a user who searches for medicine information may not want to reveal the specific drugs s/he is looking for. Additionally, an information-theoretic parameter expRatio, proposed by Xu et al [21] is employed, to measure the amount of private information exposed in the user profiles. There is a close relationship between privacy and personalization effectiveness. The lower the privacy level (the more information being provided to the PMSE server for the personalization), the better the personalization results. Thus, there is a tradeoff between them. If the user is concerned with his/her own privacy, the privacy level can be set to high to provide only limited personal information to the PMSE server. Nevertheless, the personalization effect will be less effective. On the other hand, if a user wants more accurate results according to his/her preferences, the privacy level can be set to low, such that the PMSE server can use the full user profile for the personalization process, and provide better results.

PMSE employs distance to filter the concepts in the ontology. If a concept ci+1 is a child of another concept ci in our ontology-based user profile, then c_i and c_{i+1} are connected with an edge whose distance is defined, $d(c_i, c_{i+1}) = \frac{1}{pr(c_{i+1}|c_i)}$
We aim at filtering the concepts that are minDistance close to the leaf concepts and the concept c_i will be pruned when the following condition is satisfied:

$$\frac{D(c_{i-1}, c_k)}{D(root, c_{i-1}) + D(c_{i-1}, c_k)} < minDistance \qquad (2)$$

where c_{i-1} is the direct parent of c_i and c_k is the leaf concept, which is furthest away from $c_i(argmax_{ck}\, D(c_{i+1}, c_k))$ in the ontology. $D(c_{i-1}, c_k) = d(c_{i-1}, c_i) + d(c_{i+1}, c_{i+2}) + \cdots + d(c_{k-1}, c_k)$ is the total distance from c_{i-1} to c_k, and $D(root, c_i)$ is the total distance from the root node to c_{i-1}.

The filtered user profiles are transmitted to the PMSE server. Here, $expRatio$ is employed to measure the amount of information being pruned in the filter user profiles. Note that the complete user profile is Uq,0 , while the protected user profile for the query q with minDistance = p is Uq,p . Thus, the concept entropy HC (Uq,p) of the user profiles can be computed using the following equation:

$$H_C(U_{q,p}) = -\sum_{c_i \in U_{q,p}}^{t} pr(c_i) \log pr\,(c_i) \qquad (3)$$

where c_i is any concept that exists in the user profile $H_C(U_{q,0})$
for the query q. Given $H_C(U_{q,0})$ and $H_C(U_{q,p})$ the exposed privacy expRatioq,p can be computed as:

$$expRatio_{q,p} = \frac{H_C(U_{q,p})}{H_C(U_{q,0})} \qquad (4)$$

Figure2 shows $U_{hotel,0.2}$,$U_{hotel,0.5}$,$U_{hotel,1.0}$ for the query “hotel”. When minDistance=0.2, only the very specific concept “special discount rate” is pruned from $U_{hotel,0}$. The exposed privacy expRatiohotel,0.2 is 94.6%

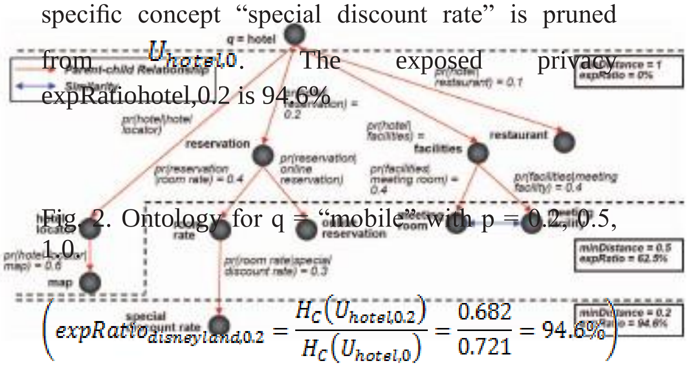
q = hotel
Parent-child Relationship
Similarity
minDistance = 1
expRatio = 0%
pr(hotel|restaurant) = 0.1
pr(hotel|reservation) = 0.2
pr(hotel|hotel locator)
pr(hotel|facilities) =
reservation
facilities
restaurant
pr(reservation|room rate) = 0.4
pr(reservation|online reservation)
pr(facilities|meeting room) = 0.4
pr(facilities|meeting facility) = 0.4
hotel locator
room rate
online reservation
meeting room
meeting facility
minDistance = 0.5
expRatio = 62.5%
pr(hotel locator|map) = 0.6
map
pr(room rate|special discount rate) = 0.3
special discount rate
minDistance = 0.2
expRatio = 94.6%

Fig. 2. Ontology for q = “mobile” with p = 0.2, 0.5, 1.0.

$$\left(expRatio_{disneyland,0.2} = \frac{H_C(U_{hotel,0.2})}{H_C(U_{hotel,0})} = \frac{0.682}{0.721} = 94.6\%\right)$$

When minDistance=0.5, four specific concepts(“room rate”, “online reservation”, “meeting room” and “meeting facility”) are pruned. Notice that “map” is not removed when minDistance=0.5, because both “map” and “hotel locator” are rare concepts with low support. Since “map” and “hotel locator” are closely related with pr(hotel locator|map)=0.6, if “hotel locator” is pruned, “map” will likely be pruned too. If both of them are pruned, the protected user profile longer determine the user’s preferences on these two concepts.

Thus, “map” is retained unless minDistance is very high (minDistance > 0.92). The exposed privacy expRatiohotel,0.5 is 62.5%

$$\left(expRatio_{disneyland,0.5} = \frac{H_C(U_{hotel,0.5})}{H_C(U_{hotel,0})} = \frac{0.451}{0.721} = 62.5\%\right)$$

Finally, when minDistance = 1.0, all concepts in the user

$$\left(expRatio_{disneyland,1.0} = \frac{H_C(U_{hotel,1.0})}{H_C(U_{hotel,0})} = \frac{0}{0.721} = 0\%\right)$$

5.Personalized Ranking Functions

Upon reception of the user’s preferences, Ranking SVM (RSVM) [10] 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, a 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, it can be treated as a point in the feature space. Using the preference pairs as the input, RSVM aims at finding a linear ranking function, which holds for as many document preference pairs as possible. An adaptive implementation, available at [3], is used in our experiments. In the following, we discuss 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, content feature vector (denoted by $\emptyset_C(q,d)$) and location feature vector denoted by $\emptyset_L(q,d)$) to represent the content and location information associated with documents. The feature vectors are extracted by taking into account the concepts existing in a documents and other related concepts in the ontology of the query. For example, if a document d_k embodies the content concept c_i and location concept l_i, the weight of component c_i in the content feature vector $\emptyset_C(q,d_k)$ of document d_k is incremented by one as defined in Equation (10), and the weight l_i in the location feature vector $\emptyset_L(q,d_k)$ is incremented by one as defined in Equation (12). The similarity and parent- child relationships of the concepts in the extracted concept 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, in our ontologies. We argue that all of the above relationships may help the users to find more related information in the same class. Therefore, we assign the pre-determined weights to related concepts. The related concepts components in content and location feature vectors are thus incremented by the weights as defined in Equation (11) and Equation (13).

The extraction of content feature vector and location feature vector are defined formally as follows.

1) Content Feature Vector

If content concepts is c_i in a web-snippet s_k , their values are incremented in the content feature vector $\emptyset_C(q, d_k)$ with the following equation: $\forall c_i \in s_k, \emptyset_c(q, d_k)[c_i] = \emptyset_C(q, d_k)[c_i] + 1$

(5)

For other content concepts c_i that are related to the con- tent concept c_i in the content ontology, they are incremented in the content feature vector $\emptyset_C(q, d_k)$ according to the following equation:

$$\forall c_i \in s_k, \emptyset_c(q, d_k)[c_j] = \emptyset_C(q, d_k)[c_j] + sim_R(c_i, c_j) + ancestor(c_i, c_j) + descendant(c_i, c_j) + sibling(c_i, c_j)$$

(6)

2) Location Feature Vector

If location concept is in a web-snippet d_k , it value is incremented in the location feature vector $\emptyset_L(q, d_k)$ with the following equation: $\forall L_i \in s_k, \emptyset_L(q, d_k)[l_i] = \emptyset_L(q, d_k)[l_i] + 1$

(7)

For other location concepts l_j that are related to the concept l_i in the location ontology, they are incremented in the location feature vector $\emptyset_L(q, d_k)$ according to the following equation.

$$\forall L_i \in s_k, \emptyset_l(q, d_k)[l_j] = \emptyset_L(q, d_k)[c_j] + ancestor(l_i, l_j) + descendant(l_i, l_j) + sibling(l_i, l_j)$$

(8)

5.2 GPS Data and Combination of Weight Vectors:

GPS locations are important information that can be useful in personalizing the search results. For example, a user may use his/her mobile device to find movies on show tn the nearby cinemas. Thus, 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 $\overrightarrow{w_{L,q,u}}$ to reflect the possible preferences. Thus, if a user has visited the GPS location l_r, the weight of the location concept $\overrightarrow{w_{L,q,u}[l_r]}$ is incremented according the following equation.

$$\forall l_r \text{ that u has visited, } \underset{w_{L,q,u}}{\longrightarrow}[l_r] = \underset{w_{L,q,u}}{\longrightarrow}[l_r] + w_{GPS(u,l,t)}$$

(9)

where $w_{GPS(u,l,t)}$ is the weight being added to the GPS location , and the number of location visited since the user visit lr (tr = 0 means the current location)3 . Hence, we assume that the location that the user has visited a long time ago is less important than the location that the user has recently visited.

The weight $w_{GPS(u,l,t)}$ being added to the $\overrightarrow{w_{L,q,u}[l_r]}$ according to the following decay equation

$$w_{GPS(l,t)} = w_{GPS\_0.e}{}^{-t_r} \qquad (10)$$

where $w_{GPS\ 0}$ is the initial weight for the decay function when tr = 0.

6. Conclusion

We proposed PMSE to extract and learn a user's content and location preferences based on the user's click through. To adapt to the user mobility, we incorporated the user's GPS locations in the personalization process. We observed that GPS locations help to improve retrieval effectiveness, especially for location queries. We also proposed two privacy parameters, *minDistance* and *expRatio*, to address privacy issues in PMSE by allowing users to control the amount of personal information exposed to the PMSE 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 click through data to further enhance the personalization effectiveness of PMSE.

References

[1]Appendix.http://www.cse.ust.hk/faculty/dlee/tkde-pmse/appendix.pdf.IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING.

[2] National geospatial. http://earth-info.nga.mil/.

[3] *svmlight*. http://svmlight.joachims.org/.

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