Ontology Based Android Mobile Search Engine

Mr. V. Dilip kumar & Mr. S. Ravi kumar

School of Computing Sciences, Hindustan University, Chennai

ABSTRACT

I propose a ontology based mobile search engine, OBMASE, that captures the users' preferences in the form of concepts by mining their click through data. Due to the importance of location information in mobile search, OBAMSE classifies these concepts into content concepts and location concepts. In addition, users' locations (positioned by GPS) are used to supplement the location concepts in OBMASE. 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. To characterize the diversity of the concepts associated with a query and their relevance's to the users need, four entropies are introduced to balance the weights between the content and location facets. Based on the client-server model, we also present a detailed architecture and design for implementation of OBAMSE. In our design, the client collects and stores locally the click through data to protect privacy, whereas heavy tasks such as concept extraction, training and re-ranking are performed at the OBAMSE server. Moreover, we address the privacy issue by restricting the information in the user profile exposed to the OBAMSE server with two privacy parameters. We prototype OBAMSE on the Google Android platform. Experimental results show that OBAMSE significantly improves the precision comparing to the baseline. Moreover, we address the privacy issue by restricting the information in the user profile exposed to the server with two privacy parameters. We prototype OBMSE on the Google Android platform. Experimental results show that OBMSE significantly improves the precision comparing to the baseline.

Index Terms—Clickthrough data, concept, location search, mobile search engine, ontology, personalization, user profiling.

1. INTRODUCTION

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. 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 click through data. Leung, 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 ontology based android mobile search engine, OBAMSE which represents different types of concepts in different ontology's. 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 click through of the query "hotel", OBAMSE can learn the user's content preference (e.g., "room rate" and "facilities") and location preferences ("Japan"). Accordingly, OBAMSE will favour results that are concerned with hotel information in Japan for future queries on "hotel". The introduction of location preferences offers OBAMSE an additional dimension for capturing a users interest and an opportunity to enhance search quality for users. With the amount of data doubling each year, more data is gathered and data mining is becoming an increasingly important tool to transform this data into information. Long process of research and product development evolved data mining.

1.1 Problem Statement

In the existing system there is only the query based searching is only available, by using this query based mobile searching it is not possible to extract the data from the search engine, to avoid this problem we are using the ranking based searching and the gaps location based searching, by using these two we can easily extract the user query from the search engine

1.2 Objective of the work

In the proposed Model, users search's on the web for query, either Area specified (or) user's location, server retrieves all the information to the user's Pc where ontology us applied. User Pc displays all the relevant keywords to the user's mobile, so that user selected the exact requirement. Ranking occurs and finally exactly mapped information is produced to the users mobile. In the user mobile we want install a APK application we want to create a login page by using this page the user will be connected to the inter mediate server from there only the user will retrieve the perfect what he want .In android mobile phone we will create the key words, by using these the user will extract data easily .We will use the UDD Algorithm to remove the duplicate links and the duplicate data from at the intermediate server.

2. RELATED WORK

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. As a result, mobile users tend to submit shorter, hence, more ambiguous queries compared to their web search counterparts.

2.1 Demerits

In the existing system there is only the query based searching is only available, by using this query based mobile searching it is not possible to extract the data from the search engine, to avoid this problem we are using the ranking based searching and the gaps location based searching, by using these two we can easily extract the user query from the search engine.

3. MODEL

In the proposed model, users search's on the web for query, either Area specified (or) user's location, server retrieves all the information to the user's Pc where ontology us applied. User Pc displays all the relevant keywords to the user's mobile, so that user selects the exact requirement. Ranking occurs and finally exactly mapped information is produced to the user's mobile. In the user mobile we want install a APK application we want to create a login page by using this page the user will be connected to the inter mediate server from there only the user will retrieve the perfect what he want .In android mobile phone we will create the key words, by using these the user will extract data easily .We will use the UDD Algorithm to remove the duplicate links and the duplicate data from at the intermediate server.

3.1 Merits

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. As a result, mobile users tend to submit shorter, hence, more ambiguous queries compared to their web search counterparts. In the proposed system the user can search the data based on the query based only but in the existing system it is based on the both query based and the location based.

4. CONSTRUCTION

4.1 MOBILE USER

An Android mobile client is an application that access a service made available by a server. The server is often (but not always) on another computer, in which case the client accesses the service by way of a network. The term was first applied to devices that were not capable of running their own stand-alone programs, but could interact with remote computers via a network.

To send the request to the server, the users have to be an registered person in the server. The user have to submit their user name password and another details to the server during the registration phase. All this information is stored in the database via server for future purpose.

4.2 MAIN SERVER

A server is a computer program running to serve the requests of other programs, the "clients". Thus, the "server" performs some computational task on behalf of "clients". The clients either run on the same computer or connect through the network.

Here the Server acts as the main resource for the client. Server is responsible for maintaining all the client information. So the server will process the

ISSN: 2278-0181
Vol. 2 Issue 4, April - 2013

user's request and get the concerned data from the database.

4.3ONTOLOGY

Ontology formally represents knowledge as a set of concepts within a domain, and the relationships among those concepts. It can be used to reason about the entities within that domain and may be used to describe the domain. Here , we are using the ontology concept to group the data as per the related domain. So that, if the user search the data, the data will displayed in domain they are requesting.

5. UDD ALGORITHM:

Here I am applying **UDD Algorithm** to avoid the duplicate values from final list the will be displayed to the user. So that we may be able to minimize the total number of links as well the duplicate values from the final result page. Unsupervised duplicate Detection (UDD) –Algorithm that uses no pre-determined training data to identify duplicates that refer to the same real-world entity.

6. ARCHITECTURE AND IMPLEMENTATION

The problem of efficient query processing in location-based search systems. A query is assigned with a query footprint that specifies the geographical area of interest to the user. Several algorithms are employed to rank the search results as a combination of a textual and a geographic score. The architecture mainly it contains OBAMSE client and the OBAMSE server .The OBAMSE client mainly fetch the data from the OBAMSE server, the OBAMSE server mainly it contains the backend server ,from there the OBAMSE fetch the data then it sends to the client. The OBAMSE server it contains s location extract and the content extractor which are connected to the serve . by using the reranking process the OBAMSE extract the data and send it to the client.The client side it contains ontology updating and the click through collector they send the data to the location feature and the content feature.

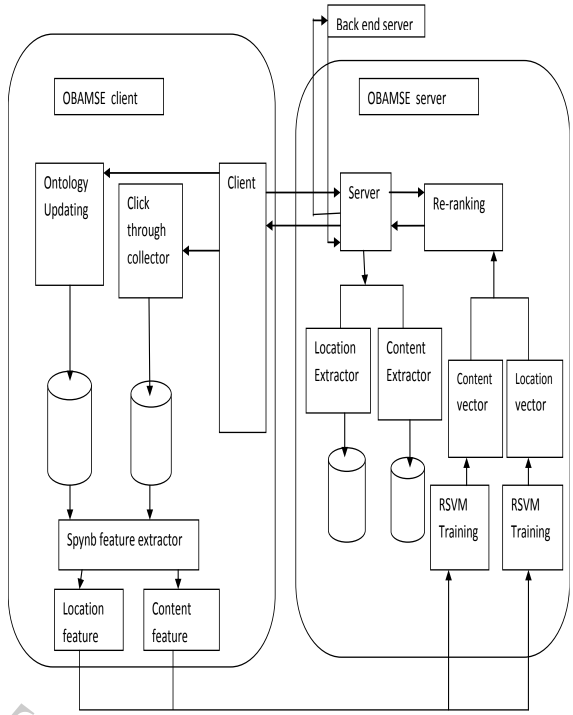

Figure 3.1 The General process flow of OBAMSE.

client-server architecture, which meets three important requirements. First, computation intensive tasks, such as RSVM training, should be handled by the OBMSE 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 OBAMSE clients in order to preserve user privacy. In the OBAMSE's client-server architecture, OBAMSE clients are responsible for storing the user clickthrough and the ontologies derived from the OBAMSE server. Simple tasks, such as updating clickthough and ontologies, creating feature vectors, and displaying reranked search results are handled by the OBAMSE clients with limited computational power. On the other hand, heavy tasks, such as RSVM training and reranking of search results, are handled by the OBAMSE server. Moreover, in order to minimize the data transmission between client and server, the OBAMSE client would only need to submit a query together with the feature vectors to the OBAMSE 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.OBAMSE's design addressed the issues: (1) limited computational power on mobile devices, and (2) data transmission minimization. OBAMSE consists of two major activities: 1) Reranking the search results at the OBAMSE server, and 2) Ontology update and clickthrough collection at a mobile client.

7. RESULTS

I proposed OBAMSEto extract and learn a user's content and location preferences based on the user's clickthrough. 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 OBAMSE by allowing users to control the amount of personal information exposed to the OBAMSE 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 OBAMSE.

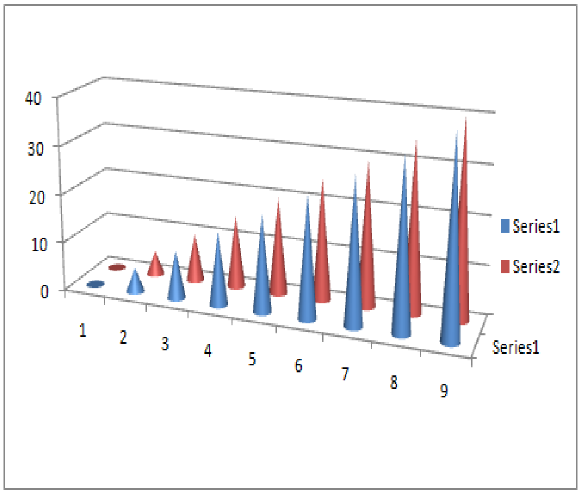

Fig 2. ARRs for (*HC*(*q*),*HL*(*q*)) query classes

CONCLUSION

We propose the ranking adaptation to adapt the well learned models from the broad-based search or any other auxiliary domains to a new target domain. By model adaptation, only a small number of samples need to be labelled, and the computational cost for the training process is greatly reduced. Based on the regularization framework, the Ranking Adaptation SVM algorithm is proposed, which performs adaptation in a black-box way. Based on RASVM, two variations called RA-SVM margin rescaling and RA-SVM slack rescaling are proposed to utilize the domain specific features to further facilitate the adaptation, by assuming that similar documents should have consistent rankings, and constraining the margin and loss of RA-SVM adaptively according to their similarities in the domain-specific feature space. Then we propose ranking adaptability, to quantitatively measure whether an auxiliary model can be adapted to a specific target domain and how much assistance it can provide.

9. FUTURE WORK

TO decrease human work on the mobiles ,feature work should evaluate more algorithms to work the application very effectively An increasing number of users are performing searches on the Web in unfamiliar domains. However, because many users lack domain-specific search knowledge, their searches are often ineffective. An important remedy is to make domain-specific search knowledge in these new domains explicit and available. Towards that goal, healthcare and online shopping experts were observed while they performed search tasks within and outside their domains of expertise. Identified domain-specific search strategies in each domain and demonstrated that such knowledge is not automatically acquired from using general-purpose search engines. These results suggest that users should benefit from Strategy Portals that provide domain-specific knowledge to perform searches in unfamiliar domains.

REFERENCES

[1] E. Agichtein, E. Brill, and S. Domains, "Improving web search ranking by incorporating user behaviour information," in Proc. of ACM SIGIR Conference, 2006, pp.667-686

[2] E. Agichtein, E. Brill, S. Dumais, and R. Ragno, "Learning user interaction models for predicting web search result preferences," in Proc. of ACM SIGIR Conference, 2006, pp.321-350

[3] G.Y. Chen, T. Suel, and A. Markowetz, "Efficient query processing in geographic web search engines," in Proc. of ACM SIGIR Conference, 2006, pp.421-462

On-Line Resources to Build a Lexicon, 1991, pp.531-540

[4] Q. Gan, J. Attenberg, A. Markowetz, and T. Suel, "Analysis of geographic queries in a search engine log," in Proc. of LocWeb Workshop, 2008, pp.126-136

[5] T. Joachims, "Optimizing search engines using click through data," in Proc. of ACM SIGKDD Conference, 2002, pp.144-162

[6] K. W. Church, W. Gale, P. Hanks, and D. Hindle, "Using statistics in lexical analysis," in Lexical Acquisition: Exploiting

[7] U. W.T. Leung, D. L. Lee, and W.C. Lee, "Personalized web search with location preferences," in Proc. of IEEE ICDE Conference, 2010, pp.208-218

[8] U. Li, Z. Li, W.C. Lee, and D. L. Lee, "A probabilistic topic-based ranking framework for location-sensitive domain information retrieval," in Proc. of ACM SIGIR Conference, 2009.

[9] M. Liu, W. S. Lee, P. S. Yu, and X. Li, "Partially supervised classification of text documents," in *Proc. of ICML Conference*, 2002.