International Journal of Emerging Engineering Research and Technology
Volume 2, Issue 7, October 2014, PP 43-50
ISSN 2349-4395 (Print) & ISSN 2349-4409 (Online)

Advanced Preferred Search Engine

Aashish Tamsya[1], Aishwarya Tirthgirikar[2], Shyam Deshmukh[3], Sumit Devkar[4]

[1,2,3,4]B.E Computer Engineering, RMD Sinhgad School Engineering, Pune, India

Abstract: *We propose a advanced preferred search engine (APSE) 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, APSE classifies these concepts into content concepts and location concepts. In addition, users' locations (positioned by GPS) are used to supplement the location concepts in APSE. The user preferences are organized in an ontology-based, multifaceted 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 relevancies to the user's 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 APSE. 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 APSE server. Moreover, we address the privacy issue by restricting the information in the user profile exposed to the APSE server with two privacy parameters. We prototype APSE on the Google Android platform. Experimental results show that APSE significantly improves the precision comparing to the baseline.*

Keywords: *APSE: Advanced Preferred Search Engine, Search engine, Google Android platform, efficient click-through searc*

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 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 an advanced preferred search engine (APSE) 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 click through of the query "hotel," APSE can learn the user's content preference (e.g., "room rate" and "facilities") and location preferences ("Japan"). The client is responsible for receiving the user's requests, submitting the requests to the APSE server, displaying the returned results, and collecting his/her clickthoughs in order to derive his/her personal preferences.

1.1. Motivation

We conjointly acknowledge that constant content or location concept could have completely different degrees of importance to different users and queries. To formally characterize the diversity of the ideas related to a question and their relevance's to the user's want, we have a tendency to introduce the notion of content and placement entropies to live the quantity of content and placement data related to a question. Similarly, to live what quantity the user is inquisitive about the content and/or location data within the results, we propose click content and placement entropies. supported these entropies, we have a tendency to develop a technique to estimate the personalization effectiveness for a specific question of a given user, which is then accustomed strike a balanced combination between the content and placement preferences. The results are re ranked per the user's content and placement preferences before returning to the shopper.

We additionally acknowledge that a similar content or location concept might have totally different degrees of importance to different users and totally different queries. To formally characterize the diversity of the ideas related to a question and their relevance's to the user's want, we have a tendency to introduce the notion of content and site entropies to live the number of content and site data related to a question. Similarly, to live what proportion the user is inquisitive about the content and/or location data within the results, we propose click content and site entropies. supported these entropies, we have a tendency to develop a technique to estimate the personalization effectiveness for a specific question of a given user, which is then accustomed strike a balanced combination between the content and site preferences.

1.2. Aim

We propose a realistic design for APSE by adopting the metasearch approach which relies on one of the commercial search engines, such as Google to perform an actual search.

1.3. Related Work

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 an advanced preferred search engine (APSE) 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. To incorporate context information revealed by user mobility, we also take into account the visited physical locations of users in the APSE. Since this information can be conveniently obtained by GPS devices, it is hence referred to as GPS locations. GPS locations play an important role in mobile web search.

1.3.1. Drawback of existing System

1. In an existing system, GPS location is in some difficulties.
2. Some obstacles in the privacy

1.4. Proposed System

We propose a realistic design for APSE by adopting the metasearch approach which relies on one of the commercial search engines, such as Google to perform an actual search. The client is responsible for receiving the user's requests, submitting the requests to the APSE server, displaying the returned results, and collecting his/her click through in order to derive his/her personal preferences.

The APSE 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 APSE clients, thus preserving privacy to the users. APSE has been prototyped with APSE clients on the Google Android platform and the APSE server on a PC server to validate the proposed ideas.

1.4.1. Advantage of Existing System

1. This paper studies the unique characteristics of content and location concepts, and provides a coherent strategy using client-server architecture to integrate them into a uniform solution for the mobile environment.
2. The proposed advanced preferred search engine 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. Our design adopts the server-client model in which user queries are forwarded to a APSE server for processing the training and reranking quickly.
4. We implement a working prototype of the APSE clients on the Google Android platform, and the APSE server on a PC to validate the proposed ideas.
5. APSE addresses the privacy issue by allowing users to control their privacy levels with two privacy parameters, minDistance and expRatio.

2. APPLICATION

1. Fraud detection is used to suspect with marked differences between current usage and user history.
2. Business and Finance it identifies the usual features of customers who buy the same product from the company.
3. DNA Analysis it provides similarity search and comparison among DNA sequences.

3. EXPLANATION OF IMPLEMENTED METHODS

3.1. Server Process

We propose a new and realistic system design for APSE. Our design adopts the server-client model in which user queries are forwarded to a APSE server for processing the training and reranking quickly. We implement a working prototype of the APSE clients on the Google Android platform, and the APSE server on a PC to validate the proposed ideas. Empirical results show that our design can efficiently handle user requests. Privacy preservation is a challenging issue in APSE, where users send their user profiles along with queries to the APSE server to obtain personalized search results. APSE 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.

3.2. RSVM Reranking Process

We conduct a comprehensive set of experiments to evaluate the performance of the proposed APSE. Empirical results show that the ontology-based user profiles can successfully capture users' content and location preferences and utilize the preferences to the rest of the paper is organized as follows: Related work is reviewed. We present the architecture and system design of APSE. We present our method for building the content and location ontologies. We introduce the notion of content and location entropies, and show how their usage in search personalization. We review the method to extract user preferences from the click through data. We discuss the Ranking SVM (RSVM) method for learning a linear weight vector (consisting both content and location features) to rank the search results.

3.3. APSE's Client-Server Design

APSE shoppers' area unit responsible for storing the user clickthroughs and therefore the ontologies derived from the APSE server. Easy tasks, such as change clickthoughs and ontologies, creating feature vectors, and displaying reranked search results area unit handled by the APSE shoppers with restricted procedure power. On the opposite hand, significant tasks, like RSVM training and reranking of search results, area unit handled by the APSE server. Moreover, so as to reduce the information transmission between consumer and server, the APSE consumer would solely get to submit a question beside the feature vectors to the APSE server, and therefore the server would automatically come back a group of reranked search results according to the preferences expressed within the feature vectors. The data transmission price is reduced, as a result of solely the essential knowledge (i.e., query, feature vectors, ontologies and search results) area unit transmitted between consumer and server during the personalization method.

APSE's style self-addressed the issues:

1. Restricted procedure power on mobile devices,
2. Knowledge transmission reduction. APSE consists of 2 major activities:

3.4. Reranking

When a user submits a question on the APSE consumer, the query together with the feature vectors containing the user's content and site preferences (i.e., filtered ontologies in keeping with the user's privacy setting) area unit forwarded to the APSE server, that successively obtains the search results from the back-end computer program (i.e., Google). The content and site ideas area unit extracted from the search results and arranged into ontologies to capture the relationships between the concepts

3.5. RSVM Vector

The feature vectors from the client area unit then employed in RSVM coaching to get a content weight vector and a location weight vector, representing the user interests supported the user's content and site preferences for the reranking. Again, the coaching method is performed on the server for its speed. The search results area unit then re ranked in keeping with the load vectors obtained from the RSVM coaching. Finally, the re ranked results and the extracted ontologies for the personalization of future queries area unit came back to the consumer.

Metaphysics update and clickthrough assortment at APSE consumer. The ontologies came back from the APSE server contain the conception house that models the relationships between the ideas extracted from the search results. they\'re hold on within the ontology information on the consumer.

Once the user clicks on a probe result, the clickthrough knowledge together with the associated content and site concepts area unit hold on within the clickthrough information on the final method flow of APSE.

Note that the ontologies hold on the consumer area unit a similar as what was extracted on the APSE server. The client, the clickthroughs area unit holds on the APSE clients, that the APSE server does not recognize the precise set of documents that the user has clicked on. This design permits user privacy to be preserved in sure degree.

Privacy parameters, minDistance and expRatio, area unit planned to manage the number of personal preferences exposed to the APSE server. If the user worries with his/her own privacy, the privacy level are often set to high in order that solely restricted personal data are going to be enclosed within the feature vectors and passed on to the APSE server for the personalization. On the opposite hand, if a user desires more correct results in keeping with his/her preferences, the privacy level are often set to low in order that the APSE server will use the complete feature vectors to maximize the personalization impact.

4. ALGORITHMS

4.1. Ranking SVM Algorithm

4.1.1. Loss Function

Let $\tau_{P(f)}$ be the Kendall's tau between expected ranking method r^* and proposed method $r_{f(q)}$, it can be proved that maximizing $\tau_{P(f)}$ helps to minimize the lower bound of the Average Precision of $r_{f(q)}$.

4.1.2. Expected Loss Function

The negative $\tau_{P(f)}$ can be selected as the loss function to minimize the lower bound of Average Precision of $r_{f(q)}$

$$L_{expected} = -\tau_{P(f)} = -\int \tau(r_{f(q)}, r^*) dPr(q, r^*)$$

where $Pr(q, r^*)$ is the statistical distribution of r^* to certain query q.

4.1.3. Empirical Loss Function

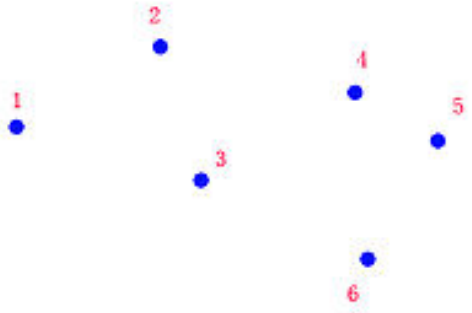
Since the expected loss function is not applicable, the following empirical loss function is selected for the training data in practice.

$$L_{empirical} = -\tau_S(f) = -\frac{1}{n}\sum_{i=1}^{n} \tau(r_{f(q_i)}, r_i^*)$$

4.1.4. Collecting Training Data

n i.i.d queries are applied to a database and each query corresponds to a ranking method. So The training data set has n elements. Each elements containing a query and the corresponding ranking method.

4.1.5. Feature Space



Labelled points in feature space

A mapping function $\Phi(q, d)$ is required to map each query and the element of database to a feature space. Then each point in the feature space is labelled with certain rank by ranking method.

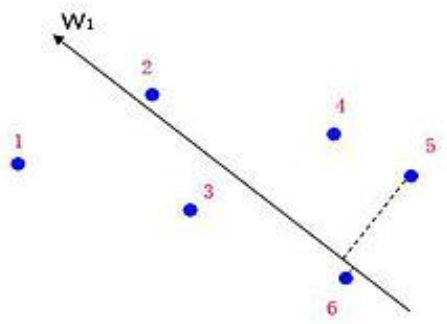
4.1.6. Optimization Problem

The points generated by the training data are in the feature space, which also carry the rank information (the labels). These labeled points can be used to find the boundary (classifier) that specifies the order of them. In the linear case, such boundary (classifier) is a vector.

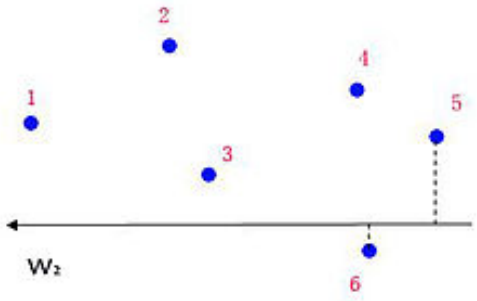
Suppose $c_i$and $c_j$are two elements in the database and denote $(c_i, c_j) \in r$ if the rank of $c_i$is higher than $c_j$in certain ranking method r. Let vector $\vec{w}$be the linear classifier candidate in the feature space. Then the ranking problem can be translated to the following SVM classification problem.Note that one ranking method corresponds to one query.

$$minimize : V(\vec{w}, \vec{\xi}) = \frac{1}{2}\vec{w} \cdot \vec{w} + C_{onstant} \sum \xi_{i,j,k}$$
$$s.t.$$
$$\forall \xi_{i,j,k} \geqq 0$$
$$\forall (c_i, c_j) \in r_k^*$$
$$\vec{w}(\Phi(q_1, c_i) - \Phi(q_1, c_j)) \geqq 1 - \xi_{i,j,1};$$
$$...$$
$$\vec{w}(\Phi(q_n, c_i) - \Phi(q_n, c_j)) \geqq 1 - \xi_{i,j,n};$$
$$where\ k \in \{1, 2, ...n\}\ ,\ i, j \in \{1, 2, ...\}\ .$$

The above optimization problem is identical to the classical SVM classification problem, which is the reason why this algorithm is called Ranking-SVM.



W candidate



Not a w candidate

4.1.7. Retrieval Function

The optimal vector $\vec{w}^*$ obtained by the training sample is

$$\vec{w}^* = \sum \alpha^*_{k,l} \Phi(q_k, c_i)$$

So the retrieval function could be formed based on such optimal classifier. For new query q, the retrieval function first projects all elements of the database to the feature space. Then it orders these feature points by the values of their inner products with the optimal vector. And the rank of each feature point is the rank of the corresponding element of database for the query q

4.2. Spy NB Method (User Preference & Privacy)

In this section, we propose a new preference mining algorithm, called Spy Naïve Bayes (SpyNB). It consists of two main components: a spying technique to obtain more accurate negative samples and a voting procedure to consider the opinions of all spies. According to our clickthrough interpretation, we need to categorize unlabeled data in order to discover the predicted negative links. Naive Bayes [Mitchell 1997] is a simple and efficient text categorization method. However, conventional Naïve Bayes requires both positive and negative examples as training data, while we only have positive examples. To address this problem, we employ a spying technique to train Naive Bayes by incorporating unlabeled training examples. Moreover, in order to obtain more accurate predicted negatives, we further introduce a voting procedure to make full use of all potential spies.

Algorithm1. *The Spy Naive Bayes (SpyNB) Algorithm*

Input. *P { a set of positive examples; U { a set of unlabeled examples; Tv { a voting threshold;*

Output.*PN { the set of predicted negative examples*

Procedure.

1: $PN_1 = PN_2 = ... = PN_{|P|} = \{\}$ and $PN = \{\}$;

2: for each example $p_i \in P$ do

3: $P_s = P - \{p_i\}$;

4: $U_s = U \cup \{ p_i \}$;

5: Assign each example in P_s the class label 1;

6: Assign each example in U_s the class label -1;

7: Train a Naive Bayes on P_s and U_s using Naïve Bayes Training algorithm;

8: Predict each example in U_s using trained Naive Bayes;

9: Spy threshold $T_s = P_r(+|p_i)$;

10: for each $u_i \in U$ do

11: if $Pr(+|u_j) < T_s$ then

12: $PN_i = PN_i \{uj\}$;

13: end if

14: end for

15: end for

16: for each $u_j \in U$ do

17: V otes = the number of PN_i such that $u_j \in PN_i$

18: if V otes > T_v . $|P|$ then

19: $PN = PN U \{u_j\}$;

20: end if

21: end for

5. INSTRUMENTATIONS

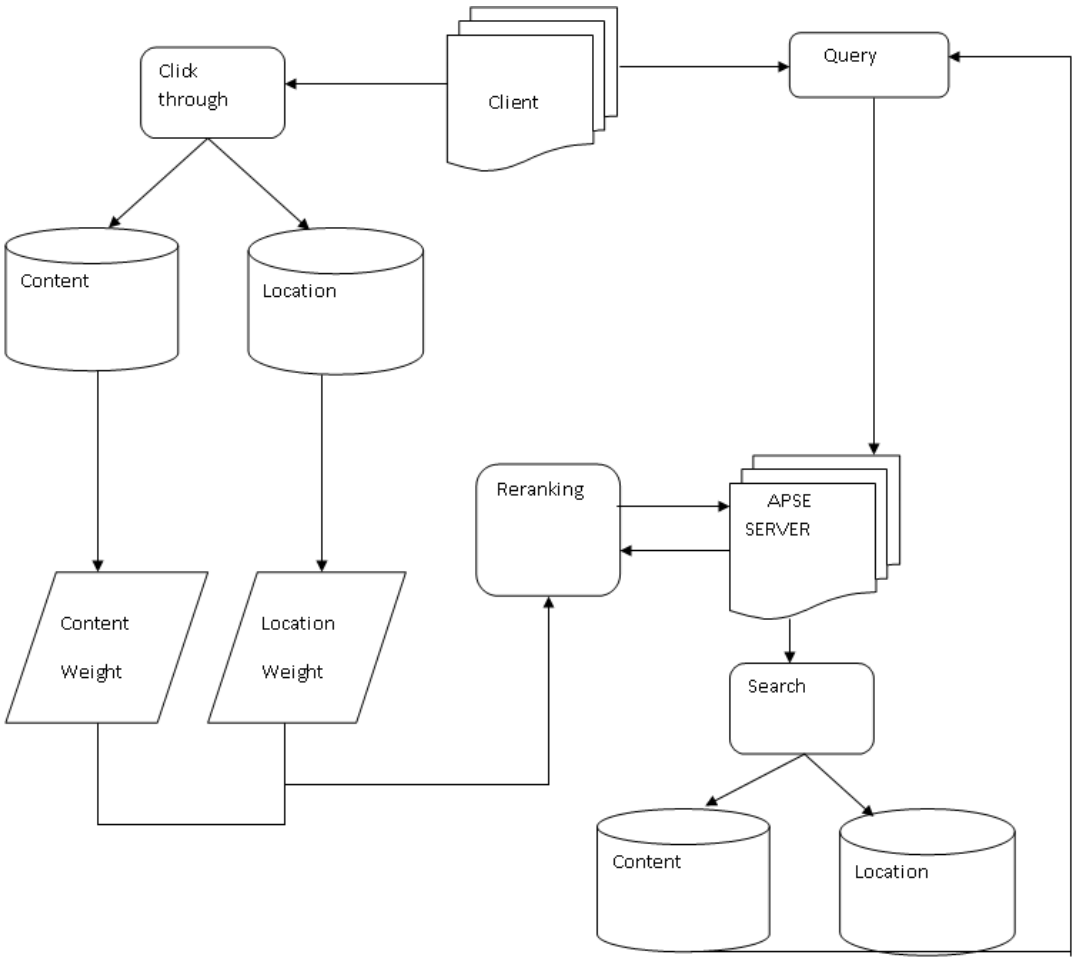


Figure 1. *System Architecture*

The user queries are stored as a clickthrough data collection in the client database. Using the clickthrough database user preference can be extracted through spyNB technique. This preference can be analyzed with the result of backend search engine and provided re-ranked search results. Thus the APSE will provide efficient search results by supporting the multiple preference of particular user. APSE maintaining good ranking quality and the data transmissions between the user and the search engine should ensure fast and efficient processing of the search. 4.2.1. The figure shows the complete architecture of the proposed system.

6. CONCLUSION

Since users are given with predefined queries and topical interests, they have to synthesize their information needs from the given queries and topical interests and conduct their searches correspondingly. Thus, their search behaviors in the experiments may be quite different from what they might have exhibited when they attempt to resolve real-life information needs. Ideally, a large-scale user study should be conducted in which APSE is subjected to real-life use, users' behaviors are monitored transparently and satisfaction of the users is analyzed and compared with other systems, but a large-scale, in-the-wild study is beyond the scope of this paper.

ACKNOWLEDGEMENT

The authors would like to express their sincere thanks to the editors and reviewers for giving very insightful and encouraging comments.

REFERENCES

[1] 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.

[2] 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.

[3] 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.

[4] C.E. Shannon, "Prediction and Entropy of Printed English," Bell Systems Technical J., vol. 30, pp. 50-64, 1951.

[5] 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.

[6] 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.

[7] E. Voorhees and D. Harman, TREC Experiment and Evaluation in Information Retrieval. MIT Press, 2005.

[8] Y. Xu, K. Wang, B. Zhang, and Z. Chen, "Privacy-Enhancing Personalized Web Search," Proc. Int'l Conf. World Wide Web (WWW), 2007.

[9] S. Yokoji, "Kokono Search: A Location Based Search Engine," Proc. Int'l Conf. World Wide Web (WWW), 2001.