International Journal of Engineering Research and Development
***e*-ISSN: 2278-067X, *p*-ISSN: 2278-800X, www.ijerd.com**
Volume 4, Issue 5 (October 2012), PP. 19-22

Personalized User Preference Mining from Weblogs by Agglomerative Concept Clustering

Dasari.Kiran Kumar, V.N.S.Vijaya Kumar, Mr. B.Suresh Kumar, Gundapu Thirupathi

Abstract:—Current web search engines are built to serve all users, independent of the needs of any individual user. Search Engine personalization is to carry out retrieval for each user incorporating his/her interests based on the user profiles. Although personalized search has been proposed for many years and many personalization strategies have been investigated, it is still unclear whether personalization is consistently effective on different queries for different users, and under different search contexts. Another major problem of current document-based web search is that search queries are usually short and ambiguous, and thus are insufficient for specifying the precise user needs. In order to address all the above problems, in this paper we are introducing an effective approach that captures the user's conceptual preferences in order to provide personalized query suggestions. To predicate the user preferences accurately we propose a new two-phase personalized agglomerative clustering algorithm that is able to generate personalized query clusters. Experimental results indicate that our technique to personalize web search is both effective and efficient.

Keywords:—concept based user profiles, search engine personalization, user preferences, agglomerative algorithm, query concept bipartite graph.

I. INTRODUCTION

As the number of Internet users and the number of accessible Web pages grows, it is becoming increasingly difficult for web search engines to find the documents that are relevant to the user needs. Now days the amount of information on the web increases rapidly, it creates many new challenges for web search. When the same query is submitted by different users, a typical search engine returns the same result, regardless of who submitted the query. This may not be suitable for users with different information needs. In fact, the vast majority of queries to search engines are short and ambiguous, and different users may have completely different information needs and goals under the same query. For example, for the query "apple", some users may be interested in documents dealing with "apple" as "fruit", while other users may want documents related to Apple computers and Apple Mac OS. One way to disambiguate the words in a query is to associate a small set of categories with the query. Due to the queries are short and ambiguous, search engines return lots of web pages as result most of them may be irrelevant to the user. The better solution to improve the user search relevance quality is personalized search. It is an important research area that aims to resolve the ambiguity of query terms. To increase the relevance of search results, personalized search engines create user profiles to capture the users' personal preferences and as such identify the actual goal of the input query. Since users are usually reluctant to explicitly provide their preferences due to the extra manual effort involved, recent research has focused on the automatic learning of user preferences from users' search histories or browsed documents and the development of personalized systems based on the learned user preferences.

In this paper, we propose a method that provides personalized query suggestions based on a personalized concept-based clustering technique. Based on the users given implicit feedback(click through data) on search results, our method will predicate the user's conceptual preferences by creating a separate profile for each user and then provides personalized query suggestions for each individual user according to his/her conceptual needs based on their profiles. All of the user profiling strategies are query-oriented, meaning that a profile is created for each of the user's queries. To create these query clusters agglomerative clustering algorithm is used in this approach. To provide the relevant docs as results to users, our approach hasthree steps. In first step, our method will do the concept extraction from web snippets.In second stepit predicts the previous users preferences based on click-through data from web search logs for that query. In third step it will creates the query-concept clusters by using agglomerative clustering algorithm. We conduct experiments to evaluate different methods and show that our concept-based two-phase clustering method yields the best precision and recall.

II. RELATEDWORK

There are several prior attempts on personalizing web search. One approach is to ask users to specify general interests. The user interests are then used to filter search results by checking content similarity between returned web pages and user interests. For example, [6] used ODP2 entries to implement personalized search based on user profiles corresponding to topic vectors from the ODP hierarchy. Unfortunately, studies have also shown that the vast majority of users are reluctant to provide any explicit feedback on search results and their interests [4]. Many later works on personalized web search focused on how to automatically learn user preferences without any user efforts. User profiles are built in the forms of user interest categories or term lists/vectors. User profiles were represented by a hierarchical category tree based on ODP and corresponding keywords associated with each category. User profiles were automatically learned from search history.

User profiling strategies can be broadly classified into two main approaches: document-based and concept-based approaches. Document-based user profiling methods aim at capturing users' clicking and browsing behaviors. Users' document preferences are first extracted from the click through data and then used to learn the user behavior model which is usually represented as a set of weighted features. On the other hand, concept-based user profiling methods aim at capturing users conceptual needs. Users browsed documents and search histories are automatically mapped into a set of topical categories. User profiles are created based on the users preferences on the extracted topical categories. Most document-based methods focus on analyzing users clicking and browsing behaviors recorded in the user click-through data. On web search engines, click through data is an important implicit feedback mechanism from users, which contains a list of ranked search results presented to the user, with identification on the results that the user has clicked on. The selected results are the documents that have been clicked by the user. Several personalized systems that employ click through data to capture users' interest have been proposed. Most concept-based methods automatically derive users' topical interests by exploring the contents of the users' browsed documents and search histories. A user profiling method based on users search history and the Open Directory Project (ODP) [6]. The user profile is represented as a set of categories, and for each category, a set of keywords with weights. The categories stored in the user profiles serve as a context to disambiguate user queries. If a profile shows that a user is interested in certain categories, the search can be narrowed down by providing suggested results according to the user's preferred categories.

III. PERSONALIZED QUERY-CONCEPT CLUSTERING AND RESULTS PREDICTION

We propose an approach that enables large-scale evaluation of personalized search. In this approach, we use click-through data recorded in query logs to predict the user requirements in web search. In general, when a user issues a query, he/she usually checks the documents in the result list from top to bottom. He/she clicks one or more documents which look more relevant to him/her, and skip the documents which he/she is not interested in. This user given information (click through data) can be used to create that user profile to know the user preference about that query. To provide the relevant docs as results to users, our approach has three steps. In first step, our method will do the concept extraction from web snippets. In second step it predicts the previous users preferences based on click-through data from web search logs for that query. In third step it will creates the query-concept clusters by using agglomerative clustering algorithm.

A. Concept extraction from web-snippets.

After a query is submitted to a search engine, a list of web-snippets are returned to the user. We assume that if a keyword/phrase exists frequently in the web-snippets of a particular query, it represents an important concept related to the query because it co-exists in close proximity with the query in the top documents. Thus, we employ the following support formula, which is inspired by the well-known problem of finding frequent item sets in data mining [7], to measure the interestingness of a particular keyword/phrase $\mathbf{c_i}$ extracted from the web-snippets arising from q: interestingness of a particular keyword/phrase $\mathbf{c_i}$ with respect to the query q:

$Support(c_i) = (sf(c_i)/n).|c_i|$

where $sf(\mathbf{c_i})$ is the snippet frequency of the keyword/phrase $\mathbf{c_i}$ (i.e. the number of web-snippets containing $\mathbf{c_i}$), **n** is the number of web-snippets returned and $|\mathbf{ci}|$ is the number of terms in the keyword/phrase $\mathbf{c_i}$. If the support of a keyword/phrase $\mathbf{c_i}$ is greater than the threshold s (s =0.03 in our experiments), we treat $\mathbf{c_i}$ as a concept for the query **q**. Table. 1 shows an example set of concepts extracted for the query "apple".Before concepts are extracted, stopwords, such as "the", "of", "we", etc., are first removed from the snippets. The maximum length of a concept is limited to seven words. These not only reduce the computational time but also avoid extracting meaningless concepts.

Table.1 Extracted concepts for query "apple" from web snippets.

Concept c_i	$support(c_i)$	Concept c_i	$support(c_i)$
mac	0.1	apple store	0.06
iPod	0.1	slashdot apple	0.04
iPhone	0.1	picture	0.04
hardware	0.09	music	0.03
mac os	0.06	apple farm	0.02

B. Predicating user preferences from click through data

To predicate the user preferences, we assume that two concepts from a query q are similar if they co-exist frequently in the web-snippets arising from the query q. According to the assumption, we apply the following well-known signal-to-noise formula from data mining [7] to establish the similarity between terms t_1 and t_2:

$$\mathbf{Sim(t_1,t_2) = log[(n.df(t_1 \cup t_2)/df(t_1).df(t_2)] / log\ n}$$

where n is the number of documents in the corpus, df(t) is the document frequency of the term t and df ($t_1 \cup t_2$) is the joint document frequency of t_1 and t_2. The similarity $sim(t_1,t_2)$ obtained using the above formula always lies between [0,1]. In the search engine context, two concepts c_i and c_j could co-exist in the following situations: 1) c_i and c_j coexist in the title, 2) c_i and c_j co-exist in the summary and 3) c_i exists in the title while c_j exists in the summary (or vice versa).Similarities for the three different cases are computed using the following formulas:

$$sim_{R,title}(c_i, c_j) = \log \frac{n \cdot sf_{title}(c_i \cup c_j)}{sf_{title}(c_i) \cdot sf_{title}(c_j)} / \log n$$

$$sim_{R,sum}(c_i, c_j) = \log \frac{n \cdot sf_{sum}(c_i \cup c_j)}{sf_{sum}(c_i) \cdot sf_{sum}(c_j)} / \log n$$

$$sim_{R,other}(c_i, c_j) = \log \frac{n \cdot sf_{other}(c_i \cup c_j)}{sf_{other}(c_i) \cdot sf_{other}(c_j)} / \log n$$

where sftitle($c_i \cup c_j$) or sfsum($c_i \cup c_j$) are the joint snippet frequencies of the concepts c_i and c_j in web-snippets' titles/summaries, sftitle(c)/sfsum(c) are the snippet frequencies of the concept c in web-snippets titles/summaries, sfother(ciUcj) is the joint snippet frequency of the concepts ci in a web snippet's title and cj in a web-snippet's summary (or vice versa). and sfother(c) is the snippet frequency of concept c in either web-snippets' titles or summaries. The following formula is used to obtain the combined similarity simR(c_i,c_j) from the three cases, where α + β + γ =1 to ensure thatsimR(c_i,c_j) lies between [0,1].

simR (c_i,c_j) = □ . simR,title (c_i,c_j) + □ .simR,summary(c_i,c_j) + □.simR,other(c_i,c_j).

C. Agglomerative clustering algorithm

We now review our personalized concept-based clustering algorithm [5] with which ambiguous queries can be classified into different query clusters. Concept-based user profiles are employed in the clustering process to achieve personalization effect. First, a query-concept bipartite graph G is constructed by the clustering algorithm with one set of nodes corresponds to the set of users' queries, and the other corresponds to the sets of extracted concepts. Each individual query submitted by each user is treated as an individual node in the bipartite graph by labeling each query with a user identifier. Concepts with interestingness weights greater than zero in the user profile are linked to the query with the corresponding interestingness weight in G. Second, a two-step personalized clustering algorithm is applied to the bipartite graph G, to obtain clusters of similar queries and similar concepts. Details of the personalized clustering algorithm. The personalized clustering algorithm iteratively merges the most similar pair of query nodes, and then the most similar pair of concept nodes, and then merges the most similar pair of query nodes, and so on.

In agglomerative clustering algorithm, which represents the same queries submitted from different users by one query node, we need to consider the same queries submitted by different users separately to achieve personalization effect. In other words, if two given queries, whether they are identical or not, mean different things to two different users, they should not be merged together because they refer to two different sets of concepts for the two users.Therefore, we treat each individual query submitted by each user as an individual vertex in the bipartite graph by labeling each query with a user identifier.After thepersonalized bipartite graph is created, our initial experiments revealed that if we apply algorithm directly on the bipartite graph, the query clusters generated will quickly merge queries from different users together, thus losing the personalization effect. We found that identical queries, though issued by different users and having different meanings, tend to have some generic concept nodes such as "information" in common.Algorithm 1 shows the details of the personalized clustering algorithm, a query-concept bipartite graph is created as input for the clustering algorithm.To implement this , we divide clustering into two steps. In the initial clustering step, an algorithm similar to BB's algorithm is employed to cluster all the queries, but it would not merge identical queries from different users. After obtaining all the clusters from the initial clustering step, the community merging step is employed to merge query clusters containing identical queries from different users.

Personalized Agglomerative Clustering Algorithm

Input: A Query-Concept Bipartite Graph G
Output: A Personalized Clustered Query-Concept Bipartite Graph Gp
// Initial Clustering
Step 1: Obtain the similarity scores in G for all possible pairs of queries using the noise-tolerant similarity function .
Step 2: Merge the pair of most similar queries (q_i,q_j) that does not contain the same queries from different users.
Step 3: Obtain the similarity scores in G for all possible pairs of concepts using the noise-tolerant similarity function.
Step 4: Merge the pair of concepts ðci;cjÞ having highest similarity score.
Step 5. Unless termination is reached, repeat steps 1-4.
// Community Merging
Step 6. Obtain the similarity scores in G for all possible pairs of queries using the noise-tolerant similarity function .
Step 7. Merge the pair of most similar queries (q_i,q_j) that contains the same queries from different users.
Step 8. Unless termination is reached, repeat steps 6 and 7.

IV. EXPERIMENTS

The query and clickthrough data for evaluation are adopted from previous work [7]. To evaluate the performance of our user profiling strategies,We used 200 test queries, which are intentionally designed to have ambiguous meanings (e.g. the query "kodak" can refer to a digital camera or a camera film). We ask human judges to determine a standard cluster for each query. The clusters obtained from the algorithms are compared against the standard clusters to check for their correctness.100 users are invited to use our search engine to search for the answers of the 200 test queries (accessible at [8]). To avoid any bias, the test queries are randomly selected from 10 different categories.The user profiles are employed by the

personalized clustering method to group similar queries together according to users' needs. The personalized clustering algorithm is a two phase algorithm which composes of the initial clustering phase to cluster queries within the scope of each user, and then thecommunity merging phase to group queries for the community.

V. CONCLUSIONS

In this paper, we proposed and evaluated personal user profiling strategies. The techniques make use of click through data to extract from web-snippets to build concept-based user profiles automatically. We applied preference mining rules to infer not only users preferences but also their community preferences, and utilized both kinds of preferences in deriving users profiles. The user profiling strategies were evaluated and compared with the personalized query clustering method that we proposed previously. Our experimental results show that profiles capturing the user preferences perform the best among the user profiling strategies.

REFERENCES

[1]. E. Agichtein, E. Brill, and S. Dumais, "Improving web search rankingby incorporating user behavior information," in Proc. of ACM SIGIRConference, 2006.

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

[3]. D. Beeferman and A. Berger, "Agglomerative clustering of a searchengine query log," in Proc. of ACM SIGKDD Conference, 2000.

[4]. S. Gauch, J. Chaffee, and A. Pretschner, "Ontology-based personalizedsearch and browsing," ACM WIAS, vol. 1, no. 3-4, pp. 219–234, 2003

[5]. T. Joachims, "Optimizing search engines using clickthrough data," inProc. of ACM SIGKDD Conference, 2002.

[6]. F. Liu, C. Yu, and W. Meng, "Personalized web search by mappinguser queries to categories," in Proc. of the International Conference onInformation and Knowledge Management (CIKM), 2002.

[7]. Open directory project. [Online]. Available: http://www.dmoz.org/

[8]. Appendix: 200 test queries. [Online]. Available: http://www.cse.ust.hk/~dlee/tkde09/Appendix.pdf

AUTHORS

Dasari.Kiran Kumar completed M.C.A from KU, Warangal, M.Tech from ANU, Guntur. I am presently working as Assistant Professor in Department of Computer Science Engineering in Vignana Bharathi Institute of Technology, Ghatkesar, Aushapur, Hyderabad. I am having 5years of Teaching Experience. My interested subjects are Software Engineering, Data mining, Web services, Web Technologies, Java, C, and C++.....

Mr. V.N.S.Vijaya Kumar is currently working as Assistant Professor in Lenora College of Engineering & Technology & Online Exams In charge. Mr. V.N.S.VIJAYA KUMAR has completed M.Tech (CSE) from JNTU, Kakinada. He has more than 6 Years of teaching Experience. His research areas are Software Engineering, Computer Networks, Data mining & Systems Programming. He has participated good Number of publications. He has conducted Research metho*dologies Workshop in Lenora Engineering Rampachodavaram in association with ISTE, MHRD AND IIT BOMBAY*.

Mr. B.Suresh Kumar completed MCA and M.Tech Computer Science Engineering, presently working as a Sr. Lecturer in AMITY University, Jaipur, and Rajasthan. Having 5 ½ years of Academic experience, researching on uncertainty with fuzzy systems. Published one international journal and Attended One international conference, wrote a book on Artificial intelligence and it is under publication process and research areas are Data Mining, Artificial intelligence.

GUNDAPUTHIRUPATHI B.Tech, M.tech working as Asst. Professor in SVS ENGINEERING COLLEGE BEEMAARAM, WARANGAL. His areas of interests are Data Mining, Computer Networks. He is having 7 years of experience in engineering stream.