THE INTERNATIONAL JOURNAL OF
SCIENCE & TECHNOLEDGE

Search Simple Queries in Concept Based Clustering

P. Madhusudhan
Student of MCA., Santhiram Engineering College, Nandyal, Kurnool (dt), Andhra Pradesh, India
S. Hussain Basha
Student of MCA., Santhiram Engineering College, Nandyal, Kurnool (dt), Andhra Pradesh, India
C. Hareesh
Student of MCA., Santhiram Engineering College, Nandyal, Kurnool (dt), Andhra Pradesh, India
D. Lakshmi Srinivasulu
Assistant Professor, Department of CSE
Santhiram Engineering College, Nandyal, Kurnool (dt), Andhra Pradesh, India

Abstract:
A major problem of current Web search is that search queries are usually short and ambiguous, and thus are insufficient for specifying the precise user needs. To alleviate this problem, some search engines suggest terms that are semantically related to the submitted queries so that users can choose from the suggestions the ones that reflect their information needs. In this paper, we introduce an effective approach that captures the user's conceptual preferences in order to provide personalized query suggestions. We achieve this goal with two new strategies. First, we develop online techniques that extract concepts from the web-snippets of the search result returned from a query and use the concepts to identify related queries for that query. Second, we propose a new two phase personalized agglomerative clustering algorithm that is able to generate personalized query clusters. To the best of the authors 'knowledge, no previous work has addressed personalization for query suggestions. To evaluate the effectiveness of our technique, a Google middleware was developed for collecting click through data to conduct experimental evaluation. Experimental results show that our approach has better precision and recall than the existing query clustering methods.

Keywords*: Click through, concept-based clustering, personalization, query clustering, search engine*

1. Introduction

AS the Web keeps expanding, the number of pages indexed in a search engine increases correspondingly With such a large volume of data, finding relevant information satisfying user needs based on simple search queries becomes an increasingly difficult task. Queries submitted by search engine users tend to be short and ambiguous. A study by Jansen et al. [20] found that the average query length on a popular search engine was only2.35 terms. These short queries are not likely to be able to precisely express what the user really needs. As a result lots of pages retrieved may be irrelevant to the user needs because of the ambiguous queries. On the other hand, users may not want to reformulate their queries using more search terms, since it imposes additional burden on them during searching. To improve user's search experience, most major commercial search engines provide query suggestions to help users formulate more effective queries. When a user submits a query, a list of terms that are semantically related to the submitted query is provided to help the user identify terms that he/she really wants, hence improving there trivial effectiveness. Yahoo's "Also Try" [6] and Google's" Searches related to" features provide related queries frorn barrowing search, while Ask Jeeves [1] suggests both more specific and more general queries to the user. Unfortunately these systems provide the same suggestions to the same query without considering users' specific interests.

In this paper, we propose a method that provides personalized query suggestions based on a personalized concept-based clustering technique. The motivation of our research is that queries submitted to a search engine may have multiple meanings. For example, depending on the user, the query "apple" may refer to a fruit, the company Apple Computer or the name of a person, and so forth. Thus, providing personalized query suggestion (e.g., users interested in "apple" as a fruit get suggestions about fruit, while users interested in "apple" as a company get suggestions about the company's products) certainly helps users formulate more effective queries according to their needs.

Our approach consists of the following four major steps. First, when a user submits a query, concepts (i.e., important terms or phrases in web-snippets) and their relations are mined online from web-snippets to build a concept relationship graph. Second, click througharecollected to predict user's conceptual preferences. Third, the concept We conduct experiments to evaluate different methods and show that our concept-based two-phase clustering method yields the best precision and recall. The rest of this paper is organized as follows: In Section 2, we compare our method with other similar approaches. We also discuss some works related to concept mining. In Section 3, we review BB's algorithm, which is also an effective technique in personalized query clustering. In Section 4, our concept mining method for extracting concepts from web-snippets is presented. In Section 5, we adapt BB's algorithm to our concept-based approach. We further extend the concept-based BB's algorithm to a personalized clustering algorithm by utilizing the user concept preference profiles. Experimental results comparing BB's algorithm with our methods are presented in Fig. 1. The general process of concept-based clustering

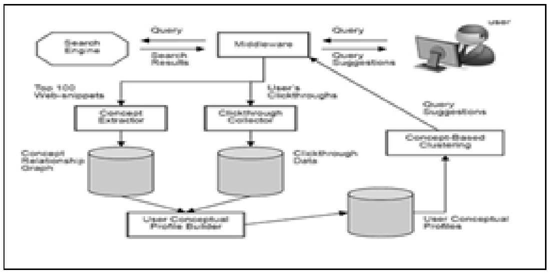


Fig. 1 shows the general process of our approach. To evaluate the performance of our approach, we developed a Google middleware for click through data collection.[2] Users were invited to test our middleware with test queries selected from a spectrum of topical categories. We evaluate the performance of our approach using the standard recall-precision measures. Beefer man and Berger's agglomerative clustering algorithm [11] (or simply called BB's algorithm in this paper) is used as the baseline to compare with our concept-based approach .Our experimental results show that the average precision at any recall level is better than the baseline method.

2. BB'S Graph-Based Clustering Algorithm

In BB's graph-based clustering [11], a query-page bipartite graph is first constructed with one set of the nodes corresponding to the set of submitted queries, and the other corresponding to the sets of clicked pages. If a user clicks on a page, a link between the query and the page is created on the bipartite graph. After obtaining the bipartite graph, an agglomerative clustering algorithm is used to discover similar queries and similar pages.

The example in Fig. 2 helps illustrate this scenario. To compute the similarity between queries or documents on a bipartite graph, the algorithm considers the overlap of their neighboring vertices as defined in the following equation:

$$sim(x,y) = \begin{cases} \frac{|N(x)\cap N(y)|}{|N(x)\cup N(y)|} & \text{if } |N(x)\cup N(y)| > 0, \\ 0, & \text{otherwise.} \end{cases} \quad (1$$

where NðxÞ is the set of neighboring vertices of x, and NðyÞ is the set of neighboring vertices of y. Intuitively, the similarity function formalizes the idea that x and y are similar if their respective neighboring vertices largely overlap and vice versa.

$$sim(x,y) = \begin{cases} \frac{|L(x,y)|}{|L(x)\cup L(y)|}, & \text{if } |L(x)\cup L(y)| > 0, \\ 0, & \text{otherwise.} \end{cases} \quad (2)$$

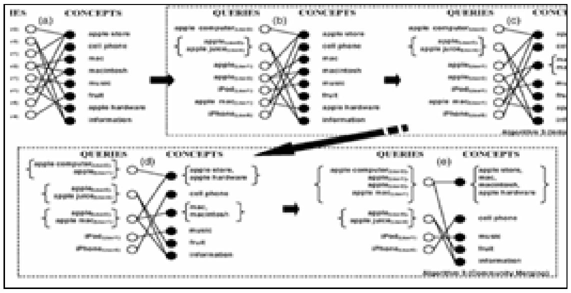
where Lðx; yÞ is the set of links connecting x and y to the same vertices, LðxÞ and LðyÞ are all the links connecting to x and y, respectively, and jLð Þj is the cardinality of Lð Þ. Applying the similarity function, we get a similarity score of 1;010=2;020 ¼ 1=2 for simðq$_1$; q$_2$ Þ in Fig. 3a, and similarity score of 1;010=3;010 ¼ 1=3 for simðq$_1$; q$_2$ Þ in Fig. 3b. Note that the score for simðq$_1$; q$_2$ Þ in Fig. 3a is higher than that in Fig. 3b, because most people are selecting document d$_1$ in Fig. 3b, and the links between q$_1$ and d$_2$ can be considered as "noise." Therefore, it is reasonable to assign a lower score to simðq$_1$; q$_2$ Þ in Fig. 3b. Using the noise-tolerant similarity function, the similarity between two vertices always lies between [0,1]. The similarity for two vertices is 0, if they share no common neighbor, and the similarity between two vertices is 1, if they have exactly the same neighbor vertices. It is noted that noise elimination by itself is a difficult problem since it requires complex inference rules to distinguish the informative from the erroneous clicks.

3. Concept-Based Clustering

Using the concepts extracted from web-snippets, we proposetwo concept-based clustering methods. We first extend BB's algorithm to a concept-based algorithm in Section 5.1. In Section 5.2, the concept-based algorithm is further enhanced to achieve effective personalized clustering. Clustering on Query-Concept Bipartite Graph We now describe our concept-based algorithm (i.e., BB's algorithm using query-concept bipartite graph) for clustering similar queries. Similar to BB's algorithm, our technique is composed of two steps: 1) Bipartite graph construction using the extracted concepts and 2) agglomerative clustering using the bipartite graph constructed in step 1.

Using the extracted concepts and click through data, the first step of our method is to construct a query-concept bipartite graph, in which one side of the vertices correspond to unique queries, and the other corresponds to unique concepts. Algorithm 1 Bipartite Graph Construction Input: Algorithm 1 Bipartite Graph Construction Input: Click through data CT , Extracted Concepts E Output: A Query-Concept Bipartite Graph G

- Obtain the set of unique queries Q ¼ fq$_1$; q$_2$; q$_3$ . . .g from CT
- Obtain the set of unique concepts C ¼ fc$_1$; c$_2$; c$_3$ . . .g from E
- Nodes ðGÞ ¼ Q [C where Q and C are the two sides in G
- If the web-snippet s retrieved using q$_i$ 2 Q is clicked by a user, create an edge e ¼ ðq$_i$; c$_j$Þ in G, where c$_j$ is a concept appearing in s.



After the bipartite graph is constructed, the agglomerative clustering algorithm is applied to obtain clusters of similar queries and similar concepts. We present the details in Algorithm 2.Algorithm 2 Agglomerative Clustering Input: A Query-Concept Bipartite Graph G Output: A Clustered Query-Concept Bipartite Graph G^c

- Obtain the similarity scores for all possible pairs of queries in G using the noise-tolerant similarity function given in (2).
- Merge the pair of queries ðq$_i$; q$_j$Þ that has the highest similarity score.
- Obtain the similarity scores for all possible pairs of concepts in G using the noise-tolerant similarity function given in (2).
- Merge the pair of concepts ðc$_i$; c$_j$Þ that has the highest similarity score.
- Unless termination is reached, repeat steps 1-4.The terminating condition for BB's algorithm is when all connected components in G^c satisfy the following conditions:

However, this terminating condition possibly generates a single big cluster of queries and a single big cluster of concepts because having the similarity threshold set to zero means that two queries (concepts) would be assigned to the same cluster even if they have only a tiny fraction of overlapping concepts (queries). To resolve this problem, we apply higher similarity thresholds, which have been observed from our experiments to yield high precision and recall:

4. Personalized Concept-Based Clustering

We now explain the essential idea of our personalized concept-based clustering algorithm with which ambiguous queries can be clustered into different query clusters. Personalized effect is achieved by manipulating the user concept preference profiles in the clustering process.

Algorithm 3 Personalized Agglomerative Clustering Input: A Query-Concept Bipartite Graph G Output: A Personalized Clustered Query-Concept Bipartite Graph G^p

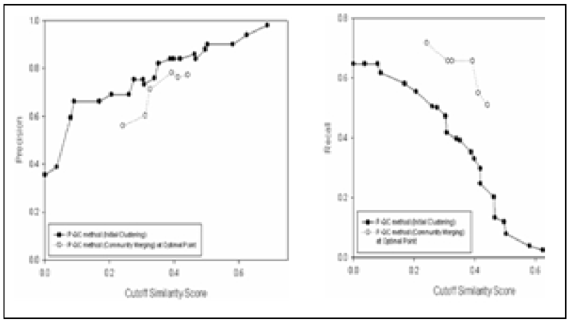
// Initial Clustering

- Obtain the similarity scores in G for all possible pairs of queries using the noise-tolerant similarity function given in (2).
- Merge the pair of most similar queries ðq$_i$; q$_j$Þ that does not contain the same queries from different users.
- Obtain the similarity scores in G for all possible pairs of concepts using the noise-tolerant similarity function given in (2).
- Merge the pair of concepts ðc$_i$; c$_j$Þ having highestsimilarity score.
- Unless termination is reached, repeat steps 1-4. // Community Merging
- Obtain the similarity scores in G for all possible pairs of queries using the noise-tolerant similarity function given in (2).
- Merge the pair of most similar queries ðq$_i$; q$_j$Þ that contains the same queries from different users.

5. Experimental Results

In this section, we evaluate the performance of the proposed clustering methods for obtaining related queries using user Click through. In Section 6.1, we first describe the experimental setup for collecting the required click through data. In Section 6.2, we compare the performance of BB's algorithm using query-URL, query-word, and query-concept bipartite graphs (or simply called the QU, QW, and QC methods). In Section 6.3, we evaluate the effectiveness of our proposed personalized concept-based clustering (or simply called the P-QC method). In Section 6.4, we discuss the algorithmic complexities based on the related parameters.

Fig.9 shows the precision-recall figures of P-QC methods. The solid line is the precision-recall graph if only initial clustering is performed. We can observe that recall is max out at 0.62. The other three lines illustrate how community merging can further improve recall beyond the limit of initial clustering. The drop of precision is due to easy merging of identical queries from different users, thus generating a single big cluster without personalization benefit When initial clustering is switched to community merging at the optimal point (see the white-circle graph in Fig. 9), community merging clearly boosts up the precision-recall envelop, which means that both precision and recall achieved in initial clustering are improved. This indicates that community merging is successful in choosing query clusters with identical queries from different users for merging.



Finally, when the switching from initial clustering to community merging is performed later than the optimal point, we can observe that recall is increased but precision is lowered, which is a typical phenomenon resulted from the conflicting nature of precision and recall. The behavior is due to the fact that overly merged clusters from initial clustering are further merged in community merging (see the dark-box graph in Fig. 9), thus further lowering the low precision generated in initial clustering. Figs. 10 and 11 show the change of precision and recall when performing P-QC method. In Fig. 10, we observe that the precisions generated by community merging are slightly lower than those generated by initial clustering because some unrelated queries can be wrongly merged in community merging. In order to further justify our choice of the parameters used in P-QC, we show in Table 10 different terminating values near the optimal point for initial clustering and community merging in the second experiment.

6. Conclusion

As search queries are ambiguous, we have studied effective methods for search engines to provide query suggestions on semantically related queries in order to help users formulate more effective queries to meet their diversified needs. In this paper, we have proposed a new personalized concept-based clustering technique that is able to obtain personalized query suggestions for individual users based on their conceptual profiles. The technique makes use of click through data and the concept relationship graph mined from web-snippets, both of which can be captured at the back end and as such do not add extra burden to users. Our experimental results confirm that our approach can successfully generate personalized query suggestions according to individual user conceptual needs. Moreover, it improves prediction accuracy and computational cost compared to BB's algorithm, which is the state-of-the-art technique of query clustering using click through for the similar objective.

There are several directions for extending the work in the future. First, instead of considering only query-concept pairs in the click through data, we can consider the relationships between users, queries, and concepts to obtain more personalized and accurate query suggestions. Second, click through data and concept relationship graphs can be directly integrated into the ranking algorithms of a search engine so that it can rank results adapted to individual users' interests.

7. References

1. Appendix, http://www.cse.ust.hk/faculty/dlee/tkde-pmse/ appendix.pdf, 2012.
2. Nat'lgeospatial,http://earth-info.nga.mil/,2012.
3. svm^{light},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 Query Processing 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