Query Graph Visualizer: A Collaborative Querying System

Lin Fu, Dion Hoe-Lian Goh, Schubert Shou-Boon Foo, Yohan Supangat

Division of Information Studies
School of Communication and Information
Nanyang Technological University
Singapore 637718
{p148934363, ashlgoh, assfoo}@ntu.edu.sg
fyohans@pmail.ntu.edu.sg

Abstract: Collaborative querying seeks to help users formulate an accurate query to a search engine by sharing expert knowledge or other users' search experiences. One approach to accomplish collaborative querying is to cluster related queries which are stored in query logs and use the related queries as recommendations to users. Here, the kernel step is to identify the similarity between queries. This paper describes a system that supports collaborative querying among its users. The system operates by clustering and recommending related queries to users using a hybrid query similarity identification approach. The system employs a graph approach to visualize the query recommendations.

1 Introduction

Collaborative querying aims to assist users in formulating queries to meet their information needs by harnessing other users' expert knowledge or search experience [CSS99] [SH04]. A common approach in collaborative querying is known as query clustering, which is to group similar queries automatically without using predetermined class descriptions. Such queries are stored in web user logs, which are then extracted and clustered to obtain recommended queries to users. A query clustering algorithm could provide a list of suggestions by offering, in response to a query Q, the other members of the cluster containing Q. In this way, there is an opportunity for a user to take advantage of previous queries and use the appropriate ones to meet his/her information need. Since similarity is fundamental to the definition of a cluster, measures of similarity between two queries are essential to the query clustering procedure. In our previous work [FGF03] [FGF04], we proposed a hybrid query similarity measure that exploits both the query terms and query results URLs. Experiments revealed that using the hybrid approach, more balanced query clusters, in terms of precision, recall, coverage and average cluster size, can be generated than using other techniques. In this paper we describe a collaborative querying system which exploits the hybrid similarity measure to cluster queries and a graph visualization approach to represent the query clusters. The system gives users the opportunity to rephrase their queries by suggesting alternate queries.

The rest of this paper is organized as follows. In Section 2, we review the literature related to this work. Next, we briefly review our approach to cluster queries and report experimental results that assess the effectiveness of this approach. We then present the query graph visualizer. A scenario is given to highlight the usefulness of this system. Finally, we discuss the implications of our work for collaborative querying systems and outline areas for further improvement.

2 Related Work

With the proliferation of online search engines and the exponential growth of information, more attention has been paid to assist the user in formulating accurate queries to express his/her information needs. A number of approaches have been proposed. One approach is to use interactive query reformulation, which aim to detect a user's "interests" through his/her submitted queries and give the user an opportunity to rephrase his/her queries by suggesting alternate queries. Several techniques have been used to incorporate aspects of interactive query reformulation systems into the information retrieval process. One technique to obtain similar queries is to use terms extracted from the search result documents. Examples include HiB [BD97], Paraphrase [AT99] and Altavista Prisma [An03], which parse the list of result documents and use the most frequently occurring terms as recommendations.

Another approach is collaborative querying. Related queries (the query clusters) may be calculated based on the similarities of the queries in the query logs [Gl01] which provide a wealth of information about past search experiences. The system can then either recommend the similar queries to users [Gl01] or use them as expansion term candidates to the original query to augment the quality of the search results [CCK90]. Here, calculating the similarity between different queries and clustering them automatically are crucial steps.

Traditional information retrieval research suggests an approach to query clustering by comparing query term vectors (content-based approach) [SM83]. Raghavan and Sever [RS95] determine similarity between queries by calculating the overlap in documents returned by the queries (results-based approach). Fitzpatrick and Dent [FD97] further developed this method by weighting the query results according to their position in the search results list. Glance [Gl01] uses the overlap of result URLs as the similarity measure instead of the document content.

3 Our Query Similarity Measure

The content-based approach might not be appropriate for query clustering since most queries submitted to search engines are quite short [Si98] [WNZ02]. Thus query terms can neither convey much information nor help to detect the semantics behind them since the same term might represent different semantic meanings, while on the other hand, different terms might refer to the same semantic meaning.

For the result URLs-based approach, the same document in the search results listings might contain several topics, and thus queries with different semantic meanings might lead to the same search results.

Thus, we hypothesize that using both query terms and the corresponding results may compensate for the drawbacks inherent in each method. The hybrid approach is a linear federation of content-based and result-based approaches. The effect of each individulal approach on the hybrid approach can be controlled by assigning a parameter to them respectively. Here, two queries are similar when (1) they contain one or more terms in common; or (2) they have results that contain one or more items in common. Further two queries are in one cluster whenever their similarity is above a certain threshold. We construct a query cluster G for each query in the query set using the definition in (1).

$$G(Q_i) = \{ Q_j : Sim(Q_i, Q_j) \geq threshold \} \quad (1)$$

where $1 < j < n$; n is the total number of query; 0=<threshold<=1.

Our previous experiments confirm our hypothesis that a combination of both query terms and result URLs provide a better overall quality of query clusters than using each separately.Here four metrics are used to measure the quality of query clusters including coverage, average cluster size, precision and recall. Compared with content-based approach, the hybrid approach improves the precision of the query clusters without sacrificing the other aspects of cluster quality significantly. Similarly, compared with result-based approach, the hybrid approach enhanced the quality of query clusters in terms of coverage, average cluster size and recall without damaging precision. More experimental results and the algorithm description can be found in [FGF03] [FGF04].

4 Query Graph Visualizer

Based on the query clusters constructed by using the hybrid query similarity measure, we developed a collaborative querying system, the Query Graph Visualizer (QGV), which displays query clusters in a graph (Figure 1). The QGV uses different colors to denote different levels from the original query. For example, yellow is used for the root node (original query) and dark blue is used for the first level (queries directly related to the root). The graph link shows the relationship between two graph nodes, with the value on the link indicating the strength of the relationship. For example, 0.1 on the link between the nodes “data mining and knowledge discovery” and “data mining journal” shows the similarity weight between these two nodes is 0.2. In addition, the system offers a tool bar to manipulate the graph visualization area including zooming, rotating and localization. The zooming function allows users to shrink or enlarge the graph visualization area. The rotating function allows users to view the visualization area from different directions. Finally, localization allows various levels of related queries to be displayed. The query graph visualizer runs as an independent agent and can be incorporated to various search engines.

By right clicking on an individual node, a popup menu appears, offering a variety of options. Firstly, users can use the selected query node and post it to a search engine (e.g. digital library at Nanyang Technological University). Secondly, users may use this query to carry out another round of searches across the query repository and detect queries related to the selected one. Further, users can expand and collapse each query node on the graph to show or hide its child nodes. Note that the label on the figure beside each node denotes how many child nodes that have not been expanded yet.

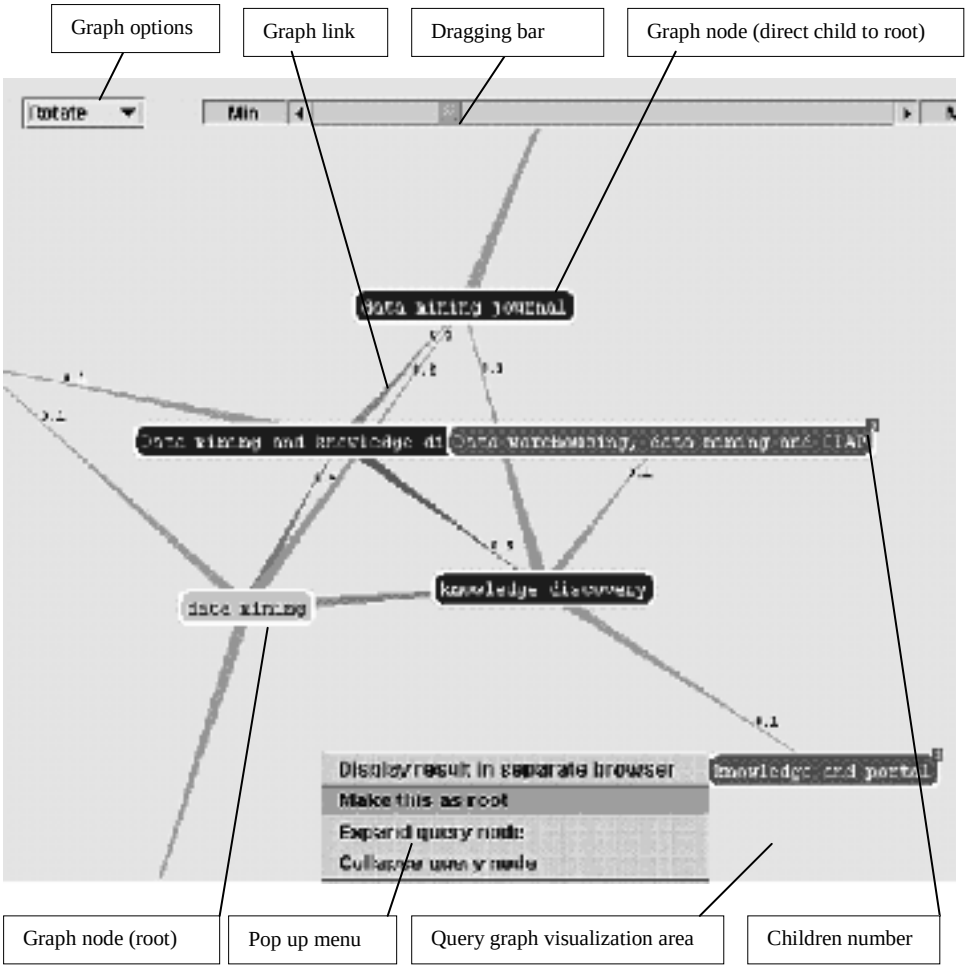

Figure 1: Query Graph Visualizer

5 A Scenario of Use

The following scenario illustrates one of the potential users of the system and highlights the operation of the system. Suppose a user is interested in the field of data mining and he is a novice in this area. When he uses the digital library system of his choice, the user first submits a query “data mining” to search for information. A moment later, a list of queries related to “data mining” is displayed as the query recommendation in addition to the search results. After looking through the result list and the recommended queries, he wants to generate a query graph using “data mining” as the root node. Thus the user triggers the QGV.

A query graph will appear on the visualization area (see Figure 1 for an example). While browsing the graph, he is interested in the node “knowledge discovery”. It is a new phrase to him but seems related to his search topic. Wanting to peruse the related queries to “knowledge discovery”, he zooms in the visualization area by dragging the bar next to the option box from left to right. He may also rotate the visualization area to facilitate his browsing by looking at the diagram from different directions. By adjusting the localization level, the user expands or collapses the nodes that contain child nodes in order to obtain an overview about the whole structure of all the queries related to the root node.

Now the user notices that there is a small number “3” near the node “data warehousing , data mining and OLAP” (see Figure1). The number here indicates that this node has three child nodes which have not been expanded. He right clicks on the node and chooses ‘Expand this node’ on the popup menu. Note this action will only take affect the selected node while the locality option discussed previously take effect across the whole visualization area. After examining the graph carefully, the user is prepared to carry out another around of information retrieval by using the node “knowledge discovery”. He thus right clicks on the node and chooses “display result in a separate browser”. The query “knowledge discovery” will be posted to the search engine automatically and the results will be displayed in a separate browser. He may repeat this process until he finds the desired information.

6 Conclusion and Future Work

In this paper, we introduced a collaborative querying system which utilizes the hybrid query similarity measure to generate query clusters for each query and employs a graph scheme to visualize the query clusters. Our work can contribute to research in collaborative querying systems that mine query logs to harness the domain knowledge and search experiences of other information seekers found in them.

In addition to the initial experiments performed in this research, alternative approaches to visualize query clusters will also be attempted, e.g., tree structure. In addition, a user evaluation to test the usefulness and usability of the collaborative querying system will be conducted.

Acknowledgement

This project is partially supported by the Nanyang Technological University (NTU) research grant RCC2/2003/SCI. Further, we would like to express our thanks to the NTU Library and the Centre for Information Technology Services at NTU for providing access to the queries.

References

[An03] Anick, P. G: Using terminological feedback for web search refinement: A log-based study. *Proceedings of SIGIR'03*, 88-95.

[AT99] Anick, P.G.; Tipirneni, S: The paraphrase search assistant: Terminological feedback for iterative information seeking. *Proceedings of SIGIR 99*, 153-161.

[BD97] Bruza, P.D.; Dennis, S: Query reformulation on the Internet: Empirical data and the Hyperindex search engine. *Proceedings of the RIAO 97*, 488-499.

[CCK90] Crouch, C.J.; Crouch, D.B.; Kareddy, K.R: The automatic generation of extended queries, *Proceedings SIGIR'90*, 269-283.

[CSS99] Churchill, E.F.; Sullivan, J.W.; Snowdon, D: Collaborative and co-operative information seeking, *CSCW'98 Workshop Report 20*(1), 1999, 56-59.

[FD97] Fitzpatrick, L.; Dent, M: Automatic feedback using past queries: Social searching? *Proceedings of SIGIR'97*, 306-313.

[FGF03] Fu, L.; Goh, D.; Foo, S: Collaborative querying through a hybrid query clustering approach. *Proceedings of ICADL'03*, 111-122.

[FGF04] Fu, L.; Goh, D.; Foo, S: Query clustering using a hybrid query similarity measure. WSEAS Transactions on Computers, *3*(3), 2004, 700-705.

[Gl01] Glance, N. S: Community search assistant. *Proceedings of IUI'01*, 91-96.

[RS95] Raghavan, V. V.; Sever, H: On the reuse of past optimal queries. *Proceedings of the SIGIR'95*, 344-350.

[SH04] Setten, M.V.; Hadidiy, F.M: Collaborative Search and Retrieval: Finding Information Together. Available at: https://doc.telin.nl/dscgi/ds.py/Get/File-8269/GigaCE-Collaborative_Search_and_Retrieval__Finding_Information_Together.pdf

[Si98] Silverstein, C. et. al.: Analysis of a very large Altavista query log. *DEC SRC Technical Note 1998-14*.

[SM83] Salton, G.; Mcgill, M.J: *Introduction to Modern Information retrieval*. McGraw-Hill New York, NY, 1983.

[WNZ02]Wen, J.R.; Nie, J.Y.; Zhang, H.J: Query clustering using user logs. *ACM Transactions on Information Systems, 20*(1), 2002, 59-81.