A Review On Query Log And Query Clustering

C.S.Dadiyala
Abha Gaikwad-Patil college of Engineering, Nagpur, India

Pragati Patil
Abha Gaikwad-Patil college of Engineering, Nagpur, India

Girish Agrawal
Abha Gaikwad-Patil college of Engineering, Nagpur, India

Abstract

User performs various complex operations over the web such as planning purchases, finance related work or researches. It's always easy to decompose the complex task or query into smaller ones to issue multiple queries. Web search engines play a vital role and keep the track of a user over long-time period. Organizing the user search histories is one of the ways to improve the output. In this paper, we study, analyse and review various techniques for organizing user search histories in terms of generation of query logs, query groups and their clustering and their potential

Keywords- *Click graph, query clustering, query grouping, search history, user history.*

1. INTRODUCTION

As the web is growing very rapidly, a user interacts very often and carries out many complex-task oriented operations over the net. The burst in the size and the richness of web is directly proportional to the variety and the complexity of task performed by user. Hence, the behaviour of a user is unpredictable and untraceable as in a user may perform many different search terms over small period of time or may perform many similar searches at different times. Query log generated by any user are hence no longer related to issuing simple navigational queries. Various studies on query logs (e.g., Yahoo's [1] and AltaVista's [2]) reveal that only about 20% of queries are navigational, while remaining are just transactional or navigational. The main reason is now user follows much elaborate task-oriented goals and operations such as planning a tour, planning a purchase & related decisions, managing their finances. The main way of accessing the information over the internet is through keywords and queries using a search engine. A search engine has become a very important component of internet and they are broadly used for accessing any information over the net. However, a user decomposes the complex task-oriented operation into number of smaller and simplified queries, such as purchasing decision can be broken down into number of co-dependent steps over a period of time. For instance, a user may first search on possible choices of mobile phones depending upon budget, manufacturing company, features, comparison among few of them, etc. After deciding which mobile phone is to be purchased, the user may search for from where to buy to get better price and post purchase services, etc. Each step requires one or more queries, and each query results in one or more clicks on relevant pages.

During their complex search online, one of the important step towards providing services and features that can help users is the capability to identify and group related queries together. This can be traced by using a new feature provided by any search engine which gives a user about their post navigational and task-oriented clicks and queries generally termed as "search histories".

In fact, identifying groups of related queries has applications beyond helping the users to make sense and keep track of queries and clicks in their search history. Hence query grouping allows the search engine to better understand the user search behaviour according to his need and his session. Once the query grouping is identified, the search engine can represent

the result of current queries and clicks by the user in the context. Query suggestions, result ranking, query alterations, sessionization, and collaborative search are the key components of search engines, which may be improved via proper query groping. For example, if a search engine knows that a current query "mobile purchase" belongs to a {"iPhone5", "mobile purchase"} query group, it can boost the rank of the page that provides information about how to get a iPhone5 instead of the Wikipedia article on "Mobile purchase" , or the pages related to mobile purchase from other mobile manufacturing companies.

Query grouping can also help different users by promoting task-level collaborative search. For example, a group of queries provided by expert users, we can select the one which is highly relevant to the current user's activity and can suggest it to him.

In this paper, we study the main concept of organizing users search histories and their various techniques. We further elaborate the problem of organizing users search histories in automated and dynamic fashion. Set of query groups is a collection of queries by the same user and related to each other over common information. The user issues new queries over time and then it is updated dynamically.

Organizing the user search histories are a challenging task for number of reasons. First, if a search task is carried over a span of time, the searched queries may or may not relate to each other over similar context. This is further complicated with multitasking such as frequently changing the search topics or using multiple tabs for different queries. Second, related queries may not be textually similar. For instance, iPhone5 and Apple are totally different in terms of textual similarity. Hence, relying only on the string similarity is not sufficient. Finally, as users may also manually alter their respective query groups, any automated query grouping has to respect the manual efforts or edits by the users. To achieve more effective and robust query grouping, we do not count on textual or temporal properties of queries.

The rest of the paper is organized as follows. In section 2, we state the goal of our paper, study and analysis of query log and generating dynamic query groups, etc. In section 3, we state how we can extract the semantic relations from the query log. In section 4, we state different approaches for query clustering. In section 5, we will see how random walk helps in our context. In section 6, we will conclude with a summary on our research

2. GOAL

Our main goal is to organize the user search histories into query groups, each containing one or more related queries and their corresponding clicks. The main objective is to analyse the query log generated by the user and then use them for further operations like, generating query group, extracting semantics relations from query log, clustering them, query expansion, etc.

2.1. QUERY LOG

As user performs the search procedure over a period of time, a query log is been generated and contains very important features. A query log contains a wealth of valuable knowledge about how web users interact with search engines as well as information about the interests and the preferences of those users. Extracting behavioural patterns from query log is a key step towards improving the service provided by search engines and towards developing innovative web search paradigms.

2.2. QUERY GROUP AND DYNAMIC QUERY GROUP

A query group is an ordered list of queries, q*i*, together with the corresponding set of clicked URLs, clk*i* of q*i*. Each query group corresponds to an atomic information need that may require a small number of queries and clicks related to the same search goal.

The process of identifying the query group is to first consider every query as a singleton query group, and then merge these singleton query groups in a iterative manner (in a k-means or agglometric way[8]).

3. EXTRACTING SEMANTIC RELATION FROM QUERY LOGS

Most of the work on query similarity is related to query expansion or query clustering. One early technique

proposed by Raghavan and Sever [14] attempts to measure query similarity using the differences in the ordering of documents retrieved in the answers, which is not feasible in the current Web. Later, Fitzpatrick and Dent [11], measured query similarity using the normalized set intersection of the top 200 documents in the answers for the queries. Again, this is not meaningful in the Web as the intersection for semantically similar queries that use different synonyms can and will be very small. Wen et al [17] proposed to cluster similar queries to recommend URLs to frequently asked queries of a search engine.[7]

They used four notions of query distance based on: (1) keywords or phrases of the query; (2) string matching of keywords; (3) common clicked URL's; and (4) the distance of the clicked documents in some pre-defined hierarchy. Befferman and Berger [4] also proposed a query clustering technique based on distance notion (3). As the average number of words in queries is small (about two) and the number of clicks in the answer pages is also small, notions (1) an (2) generate very sparse distance matrices. Notion (4) needs concept taxonomy and the clicked documents to be classified into the taxonomy, which cannot be done in a large scale. Also (3) is sparse, but this sparsity can be diminished using large query logs. The query log is viewed as a set of transactions, with each transaction representing a session in which a single user submits a sequence of related queries in a time interval. The method shows good results, but two problems arise: it is difficult to determine sessions of queries belonging to the same search process; moreover the most interesting related queries, those submitted by different users, cannot be discovered, since the support of a rule increases only if its queries appear in the same query session (i.e. they are submitted by the same user.) Baeza-Yates et al. [4, 6] used the content of clicked Web pages to define a term-weight vector model for a query. They consider terms in the URLs clicked after a query. Each term is weighted according to the number of occurrences of the query and the number of clicks of the documents in which the term appears. Then the similarity of two queries is equivalent to the similarity of their vector representations, like the cosine distance function. This notion of query similarity has several advantages. First, it is simple and easy to compute. On the other hand, it allows relating queries that happen to be worded differently but stem from the same topic, hence capturing semantic relationships among queries.Recently, Sahami and Heilman [15] used a query similarity based on the snippets of the answers to the queries. However, they do not consider the feedback of the users (i.e. clicked pages)[7,18].

4. QUERY CLUSTERING

Query clustering is a process used to find frequently searched or popular topics on a search engine. This process is crucial for search engines due to the short lengths of queries; approaches based on keywords are not suitable for query clustering. A new query clustering method that makes use of user logs which allow us to identify the documents the users have selected for a query. The similarity between two queries may be deduced from the common documents the users selected for them. A combination of both keywords and user logs is better than using either method alone.[7]

Although the need for query clustering is relatively new, there have been extensive studies on document clustering, which is similar to query clustering. In this section, we give a review of some approaches related to query clustering.

4.1. Using Keywords

In this approach, a document is represented as a vector in a vector space formed by all the keywords. Researchers have been concerned mainly with the following two aspects: (1) similarity function (2) algorithms for the clustering process.

Keyword-based document clustering has provided interesting results. One contributing factor is the large number of keywords contained in documents. Even if some of the keywords of two similar documents are different, there are still many others that can make the documents similar in the similarity calculation. However, since queries, especially the queries submitted to the search engines, typically are very short, in many cases it is hard to deduce the semantics from the queries themselves. Therefore, keywords alone do not provide a reliable basis for clustering queries effectively.

In addition, words such as “where” and “who” are treated as stop words in traditional IR methods. For questions, however, these words (if they occur) encode important information about the user’s need, particularly in the new-generation search engines such as AskJeeves. For example, with a “who” question, the user intends to find information about a person. So even if a keyword-based approach is used in query clustering, it should be modified from that used in traditional document clustering.

4.2. Using Hyperlinks

Because of the limitations of keywords, people have been looking for additional criteria for document clustering. One of them is the hyperlinks between documents. The hypothesis is that hyperlinks connect similar document. More recent examples are Google (http://www.google.com) and the authority/hub calculation of Kleinberg [1998]. Although Google does not perform document clustering explicitly, its PageRank algorithm still results in a weighting of hyperlinks. For a document, it is then straightforward to know the documents that are the most strongly related to it according to the weights of the hyperlinks to/from the document. Therefore, we can see PageRank as an implicit clustering approach. Google’s use of hyperlinks has been very successful, making it one of the best search engines currently available.

4.3. Using Cross-reference between Queries and Documents

By cross-reference, we mean any relationship created between a query and a document. The intuition of using cross-references is that similarity between documents can be transferred to queries through these references, and vice versa.[7]

5. QUERY CLUSTERING USING USER LOGS

5.1. Clustering Principles

The approach is based on two criteria: one is on the queries themselves, and the other on cross-references. We formulate them as the following principles:

Principle 1 (*using query contents*): If two queries contain the same or similar terms, they denote the same or similar information needs. Obviously, the longer the queries, the more reliable is principle 1. However, as queries are short, this principle alone is not sufficient. Therefore, the second criterion is used as a complement.[13]

Principle 2 (*using document clicks*): Two queries are similar if they lead to the selection of the same or similar document. Document selections (or document clicks) are comparable to user relevance feedback in a traditional IR environment, except that document clicks denote implicit and not always valid relevance judgments.[13]

The two criteria have their own advantages. In using the first criterion, we can group together queries of similar compositions. In using the second criterion, we benefit from user’s judgments.

6. QUERY CLUSTERING BASED ON SIMILARITY OF TEXT SNIPPETS

In analyzing text, there are many situations in which we wish to determine how similar two short text snippets are. For example, there may be different ways to describe some concept or individual, such as “United Nations Secretary-General" and “Koff Annan", and we would like to determine that there is a high degree of semantic similarity between these two text snippets. Similarly, the snippets “AI" and “Artificial Intelligence" are very similar with regard to their meaning, even though they may not share any actual terms in common.[12]

To address this problem, we would like to have a method for measuring the similarity between such short text snippets that captures more of the semantic context of the snippets rather than simply measuring their term-wise similarity. To help us achieve this goal, we can leverage the large volume of documents on the web to determine greater context for a short text snippet [12]. By examining documents that contain the text snippet terms we can discover other contextual terms that help to provide a greater context for the original snippet and potentially resolve ambiguity in the use of terms with multiple meanings.

The similarity function is based on query expansion techniques, which have long been used in the

Information Retrieval community. Such methods automatically augment a user query with additional terms based on documents that are retrieved in response to the initial user query or by using an available thesaurus. Our motivation for and usage of query expansion greatly differs from this previous work, however. First, the traditional goal of query expansion has been to improve recall (potentially at the expense of precision) in a retrieval task [12]. Our focus, however, is on using such expansions to provide a richer representation for a short text in order to potentially compare it robustly with other short texts. Moreover, traditional expansion is focused on creating a new query for retrieval rather than doing pair-wise comparisons between short texts.

7. RANDOM WALK

A search engine can track which of its search results were clicked for which query. For a popular system, these click records can amount to millions of query-document pairs per day. Each pair can be viewed as a weak indication of relevance: that the user decided to at least view the document, based on its description in the search results [16].

We can use the clicks of past users to improve the current search results. However, the clicked set of documents is likely to differ from the current user's relevant set. Some differences arise because we are aggregating clicks across users, who may simply disagree about which documents are relevant. Other differences are due to presentation issues; for example, the user must decide whether to click based on a short summary and is influenced by the ordering of results [3]. For any given search, a large number of documents are never seen by the user, therefore not clicked.

7.1. Random walk concept

To derive our probabilistic retrieval model, we first propose a basic query formulation model. The model captures a process that starts from an information need and ends with a query. We assume that query formulation begins with the use imagining a single document, representing their information need. They then think of a query that is associated with the document. The process might stop at that query, at which point they issue the query. Alternatively, the query makes them imagine another document, and that document makes them imagine another query. This thought process of query-document and document-query transition can repeat, or it can stop at a query which is then issued.

This model makes a number of simplifying assumptions. The user has limited memory, so forgets their previous location after each transition. Although they do not remember their starting point, our model limits the number of transitions to keep them in the vicinity of their information need. We do not base our model on a real study of query formulation behaviour, but instead estimate our transition probabilities from clicks of many users. It is also a simplifying assumption to use a single document to represent the information need.

7.2. Walk parameters

The behaviour of the Markov random walk is affected by the transition matrix and the number of steps in the walk. The number of steps determines the resolution of the walk. A short walk preserves information about the starting node at a fine scale; start nodes close to the end node have much higher probability than the others, and nodes further away cannot even be reached and have zero probability. A long walk preserves only coarse information about what cluster of nodes the walk was started from.

8. CONCLUSION

We described the process of generation of query log by a user using a web search engine accessing any information over period of time. This query log can be grouped and then this query logs are used to extract semantics relations. We also described that there are two different techniques for query clustering, such as using user logs and using similarity of text snippets. But both the techniques are having some disadvantages over one another. We further explained the concept of random walk. Here we studied the basic concepts about organizing a user search histories for better performance.

REFERENCES

[1] J. Teevan, E. Adar, R. Jones, and M. A. S. Potts, "Information reretrieval: repeat queries in yahoo's logs," in SIGIR. New York, NY, USA: ACM, 2007, pp. 151–158.

[2] A. Broder, "A taxonomy of web search," SIGIR Forum, vol. 36, no. 2, pp. 3–10, 2002.

[3] R. Baeza-Yates, C. Hurtado, M. Mendoza, and G. Dupret. Modeling user search behavior. In LA-WEB '05: Proceedings of the Third Latin American Web Congress, page 242, Washington, DC, USA, 2005. IEEE Computer Society.

[4] R. Baeza-Yates, C. Hurtado, and M. Mendoza. Query clustering for boosting web page ranking. AWIC'04,

[5] A. Cid, C- Hurtado, and M- Mendoza. Automatic maintenance of Web directories using clickthrough data. WIRI'06.

[6] R. Baeza-Yates, C. Hurtado, and M. Mendoza. Query recommendation using query logs in a search engine. EDBT Workshops, 2004.

[7] R. Baeza-Yates and A. Tiberi, "Extracting Semantic Relations from query Logs," Proc. 13th ACM SIGKDD Int'l Conf. Knowledge Discovery and Data Mining (KDD), 2007.

[8] J. Han and M. Kamber, Data Mining: Concepts and Techniques. Morgan Kaufmann, 2000.

[9] P.-J. Cheng, C.-H. Tsai, C.-M. Hung, and L.-F. Chien. Query Taxonomy Generation for Web Search (poster). CIKM'06.

[10] G. Dupret and M. Mendoza. Automatic Query Recommendation using Click-Through Data. IFIP PPAI'06.

[11] L. Fitzpatrick and M. Dent. Automatic feedback using past queries: Social searching? In SIGIR'97.

[12] M. Sahami and T.D. Heilman, "A Web-based Kernel Function for Measuring the Similarity of Short Text Snippets," Proc. the 15th Int'l Conf. World Wide Web (WWW '06), pp. 377-386, 2006.

[13] J.-R. Wen, J.-Y. Nie, and H.-J. Zhang, "Query Clustering Using User Logs," ACM Trans. in Information Systems, vol. 20, no. 1, pp. 59-81, 2002.

[14] V. V. Raghavan and H. Sever. On the reuse of past optimal queries. SIGIR'95.

[15] M. Sahami and T. D. Heilman. A web-based kernel function for measuring the similarity of short text snippets. WWW'06.

[16] N. Craswell and M. Szummer, "Random Walks on the click Graph," Proc. 30th Ann. Int'l ACM SIGIR Conf. Research and Development in Information Retrieval (SIGIR '07), 2007.

[17] J. Wen, J. Mie, and H. Zhang. Clustering user queries of a search engine. WWW'01.

[18] H. -J. Zeng, Q. -C. He, Z. Chen, W. -Y. Ma, and J. Ma. Learning To Cluster Search Results. SIGIR'04.