The effect of similarity measures on the quality of query clusters

Fu, Lin; Goh, Dion Hoe-Lian; Foo, Schubert

2004

Fu, L., Goh, D., & Foo, S. (2004). The effect of similarity measures on the quality of query clusters. Journal of Information Science, 30(5), 396-407.

https://hdl.handle.net/10356/91768

https://doi.org/10.1177/0165551504046722

The Effect of Similarity Measures on the Quality of Query Clusters

Fu, L., **Goh, D.H.**, and Foo, S. (2004). The effect of similarity measures on the quality of query clusters. *Journal of Information Science*, 30(5), 396-407.

The Effect of Similarity Measures on the Quality of Query Clusters

Lin Fu

Division of Information Studies, School of Communication and Information, Nanyang Technological University, Singapore 637718

Dion Hoe-Lian Goh

Division of Information Studies, School of Communication and Information, Nanyang Technological University, Singapore 637718

Schubert Shou-Boon Foo

Division of Information Studies, School of Communication and Information, Nanyang Technological University, Singapore 637718

Correspondence to: Fu Lin, Division of Information Studies, School of Communication and Information, Nanyang Technological University, Singapore 637718. E-mail: p148934363@ntu.etu.sg

Abstract.

Query clustering is a process that can be used to discover common interests of online information seekers and to exploit their collective search experience for the benefit of others. Harnessing such search experiences facilitates collaborative querying that in turn may help users of digital libraries and other information systems to better meet their information needs. Since similarity is fundamental to the definition of a cluster, measures of similarity between two queries is essential to the query clustering procedure. In this paper, we examine the effectiveness of different similarity measures. A set of experiments was carried out to study the impact of different similarity measures on the quality of query clusters. The results show that different similarity measures outperform each other in different query cluster quality criteria. Implications for these findings are discussed.

Keywords: Information retrieval; Collaborative querying; Query mining; Query clustering; Similarity measures.

1. Introduction

With the increasing popularity of the Internet, people have now come to depend more on the Web or digital libraries to search for information. However the performance of existing search engines is far from people's satisfaction, exacerbated by the fact that not all results returned by search engines are relevant nor of acceptable quality to information seekers. This has thus led to a situation where users are swamped with too much information, resulting in difficulty sifting through documents in search of relevant content.

The study of information seeking behaviour has revealed that interaction and collaboration with other people is an important part in the process of information seeking and use [1, 2, 3]. Given this idea, collaborative search aims to support collaboration among people when they search information on the Web or in digital libraries [4]. Work in collaborative search falls into several major categories including collaborative browsing, collaborative filtering and collaborative querying [5]. In particular, collaborative querying seeks to help users express their information needs properly in the form of a question to information professionals, or formulate an accurate query to a search engine by sharing expert knowledge or other users' search experiences with each other [5]. Query mining is one of the common techniques used to support collaborative querying. It allows users to utilize other users' search experiences or domain knowledge by analyzing the information stored in query logs (query analysis), grouping (query clustering) and extracting useful related information on a given query. The extracted information can then be used as recommendation items (used in query recommending systems) or sources for automatic query expansion. An example involving query clustering is given below.

Consider a scenario where user A is interested in peer-to-peer software. She wants to look for a particular software named Kazza but she cannot remember its name. Hence she enters "peer to peer" as the query to the search engine and obtains a list of results. However nothing in the top 50 results contains the desired information and she does not know how to modify her query. At the same time, another user B may remember the software's name and knows that good search results can be obtained by using "Kazza" as the query. In this case, B's search history is usually stored as part of the search engine's query logs. Different search engines have query logs in different formats although most contain similar information such as a session ID, address of user, submitted query, etc. Thus, by mining the information stored in such query logs, clustering similar queries and then recommending these clusters to users, there is an opportunity for user A to take advantage of previous queries and use the appropriate ones to meet her information need.

From this example, we can see that the query clustering is one of crucial steps in query mining and the challenge here is to identify the similarities between different queries stored in the query logs. The classical method in

information retrieval suggests a similarity measure between queries according to the terms used (content-based approach) [6]. Here, queries will be grouped into the same cluster if they contain one or more common terms. An alternative approach is to use the results (e.g. result URLs in Web search engines) to queries as the criteria to identify similar queries (results-based approach) [4, 7]. In this case, the query clusters are constructed by calculating the overlap between the result URLs in response to different queries.

Although much work has been done in query clustering research, there is little rigorous analysis of the performance of different query similarity measures. In other words, the effect of different query similarity measures on the quality of query clusters has not been studied to date. In this paper, a comprehensive evaluation on different query similarity measures is reported.

This work will benefit information retrieval systems, digital libraries and other information systems in better meeting the information needs of users through collaborative querying. Specifically, this work reveals the drawbacks and advantages of different query similarity measures and shed light on improving the performance of the algorithms adopted by collaborative querying systems to identify high-quality query clusters given a submitted query.

The remainder of this paper is organized as follows. In Section 2, we review the literature related to this work. Next, we describe the query similarity measures adopted in the research and the procedure used to cluster queries. We then describe the design of our evaluation experiments and report the results that assess the effectiveness of the different similarity measures. Finally, we discuss the implications of our findings for collaborative querying systems and outline areas for further improvement.

2. Related work

There are several useful strands of literature that bear some relevance to this work. This section reviews literature from these fields. Firstly, a survey of information seeking behaviour is provided as the background for this research. Next, various approaches to support collaborative search are described to address the requirement and importance of this work. Finally, a review of different query clustering approaches, the focus of this work, is presented.

2.1. Information seeking behaviour

Information seeking is a broad term encompassing the ways individuals articulate their information needs, seek, evaluate, select and use information [1]. In other words, information seeking behaviour is a purposive seeking for information as a consequence of a need to satisfy some goal. In the course of seeking, the individual may interact with people, manual information systems (such as newspapers or libraries), or with computer-based information

systems such as the World Wide Web [8]. Many researchers have worked in this area during the past several decades. Despite the differences between various models, they share a similarity—interaction and collaboration with others is a key component in the process of information seeking and use.

For example, Taylor [3] developed a model of information seeking in libraries beginning from how people articulate a question to a librarian and the ensuing negotiation process with the librarian in order to find the needed information (question-negotiation). Taylor's research demonstrates that interaction and collaboration with librarians and colleagues is a very important step during the information seeking process. Stated differently, how one harnesses other people's knowledge is an essential factor that will determine the outcome of the information seeking process.

Dervin and Dewdney's [9] Sense-Making Model reinforces Taylor's work and focuses on how individuals use the observations of others to construct pictures of reality and use these pictures to guide their search behaviour. The term "sense-making" is a label for a coherent set of concepts and methods to describe how people construct sense of their world. Thus sense-making behaviour is communicating behaviour, and information seeking and use is central to sense-making. People communicate and collaborate with others within a certain context in order to meet their own information needs and then make use of the retrieved information for different purposes.

Further, Ellis' [10] research proposed a pattern of information-seeking behaviour that included eight generic features or research activities: starting, chaining, browsing, differentiating, monitoring, extracting, verifying and ending. Typically, the starting stage includes activities characteristic of the initial search for information, for example, identifying references. This stage is often accomplished by asking colleagues or consulting literature reviews, indexes and abstracts. Ellis argues that communication with other people is a key component in the initial search for information.

2.2. Collaborative search

As described previously, collaborative search is an emerging research area that seeks to support cooperation among individuals when they search information online. Collaborative search may be divided into three types according to the ways that users search for information: collaborative browsing, collaborative querying and collaborative filtering [5]. Collaborative browsing can be seen as an extension of Web browsing. Traditional Web browsing is characterized by distributed, isolated users with low interactions between them while collaborative browsing is performed by groups of users who have a mutual consciousness of the group presence and interact with each other during the browsing process [11]. In other words, collaborative browsing aims to offer document access to a group of users where they can communicate through synchronous communication tools [12]. Examples of collaborative browsing applications include Let's Browse [11], a system for co-located collaborative browsing using user interests, and WebEx [13], a meeting system that allows distributed users to browse Web pages.

Collaborative filtering is a technique for recommending items to a user based on similarities between the past behaviour of the user and that of likeminded people [14]. It assumes that human preferences are correlated and thus if a group of likeminded users prefer an item, then the present user may also prefer it. Collaborative filtering is a beneficial tool in that it harnesses the community for knowledge sharing and is able to select high quality and relevant items from a large information stream [15]. Examples of collaborative filtering applications include Tapestry [15], a system that can filter information according to other users' annotations; GroupLens [16], a recommender system using user ratings of documents read; and PHOAKS [17], a system that recommends items by using newsgroup messages.

Collaborative querying on the other hand, assists users in formulating queries to meet their information needs by utilizing other people's expert knowledge or search experience. There are generally two approaches used. Online live reference services are one such approach, and it refers to a network of expertise, intermediation and resources placed at the disposal of someone seeking answers in an online environment [18]. An example is the Interactive Reference Service at the University of California at Irvine, which offers a video reference service that links librarians at the reference desk at the university's Science Library and students working one and a half miles away in a College of Medicine computer lab [19].

Although online live reference services attempt to build a virtual environment to facilitate communication and collaboration, the typical usage scenario involves many users depending only on several "smart librarians". This approach inherently has the limitation of overloading especially if too many users ask questions at the same time. In such cases, users may experience poor service such as long waiting times or answers that are inadequate. Further, phone, e-mail and chat, which are the common techniques, adopted by online live reference services, usually limit the librarian and patron to one-on-one communication, making the sharing of reference interviews more difficult [20].

An alternative approach is to mine the query logs of search engines and use these queries as resources for meeting a user's information needs. Historical query logs provide a wealth of information about past search experiences. This method thus tries to detect a user's "interests" through his/her submitted queries and locate similar queries (the query clusters) based on the similarities of the queries in the query logs [4]. The system can then either recommend the similar queries to users (query recommending systems) [4] or use them as expansion term candidates to the original query to augment the quality of the search results (automatic query expansion systems) [7, 21]. Such an approach overcomes the limitation of human involvement and network overloading inherent in online live reference services. Further, the required steps can be performed automatically. Here, calculating the similarity between different queries and clustering them automatically are crucial steps. A clustering algorithm could provide a list of suggestions by offering, in response to a query q, the other members of the cluster containing q. There are some commercial search engines (e.g. Lycos) that give users the opportunity to rephrase their queries by suggesting alternate queries.

2.3. Query clustering

In this section we review different techniques used to cluster queries. Three approaches including content-based, feedback-based and result-based approaches will be described here.

2.3.1 Content-based approaches

Traditional information retrieval research suggests an approach to query clustering by comparing query term vectors. In other words, common terms can be used to characterize the cluster of queries. This can be done by simply calculating the overlap of identical terms between queries. Further, various similarity measures incorporating term weights are available including cosine similarity, Jaccard similarity, and Dice similarity [6]. These measures have provided good results in document clustering due to the relatively large number of terms contained in documents. Such methods are also relatively simple and straightforward to implement for query clustering. However, the content-based method might not be appropriate for query clustering since most queries submitted to search engines are quite short [22]. A recent study on a billion-entry set of queries to AltaVista has shown that more than 85% queries contain less than three terms and the average length of queries is 2.35 [23]. Thus query terms are not able to convey much information or 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 [7].

2.3.2 Feedback-based approaches

Another approach to clustering queries is to utilize a user's selections on the search result listings as the similarity measure [22]. This method analyzes the query session logs which contain the query terms and the corresponding documents users clicked on. It assumes that two queries are similar if they lead to the selection of a similar document. Users' feedback is employed as the contextual information to queries and has been demonstrated to be quite useful in clustering queries. However the drawback is that it may be unreliable if users select too many irrelevant documents [22]. Further, the performance of such methods will be affected greatly by the lack of common documents clicked by users [24]. In other words, if users click on different documents for identical or similar queries, this method will not generate effective query clusters.

2.3.3 Result-based approaches

Raghavan and Sever [7] determine similarity between queries by calculating the overlap in documents returned by the queries. This is done by converting the query result documents into term frequency vectors. The similarity between two queries is then decided by comparing these vectors. Fitzpatrick and Dent [25] further developed this method by weighting the query results according to their position in the result list. They argue that the beginning of a result list is more likely to include a relevant document to the original query. They derive the probabilities of different result ranges to contain relevant documents empirically and use these probabilities as weights of the

query results. Using the corresponding query results is useful in boosting the performance of query clustering in terms of precision and recall [7, 25]. However this method is time consuming to perform and is not suitable for online search systems [25]. Glance [4] thus uses the overlap of result URLs as the similarity measure instead of the document content. Queries are posted to a reference search engine and the similarity between two queries is measured using the number of common URLs in the top 50 result list returned from the reference search engine.

3. Query similarity measures

This section provides definitions of different query similarity measures used in the literature as well as our evaluation experiments. Further, our method of constructing query clusters based on different query similarity measures is presented.

3.1. The content-based similarity measure

We borrow concepts from information retrieval [6] and define a set of queries as $Q=\{Q_1, Q_2 \ldots Q_i, Q_j \ldots . Q_n\}$. A single query Q_j is converted to a term and weight vector shown in (1), where q_i is an index term of Q_j and w_{iQj} represents the weight of the i^{th} term in query Q_j. In order to compute the term weight, we define the term frequency, tf_{iQj}, as the number of occurrences of term i in query Q_j and the query frequency, qf_i, as the number of queries in a collection of *n* queries that contains the term i. High term frequency indicates that a term is highly related to a query and thus more "important" in the clustering process. High query frequency, on the other hand, indicates that a term is too general to be useful as a descriptor and will not convey useful information for query clustering. Next, the inverse query frequency, iqf_i, is expressed as (2), in which n represents the total number of queries in the query collection. We then compute w_{iQj} based on (3):

$$Q_j = \{<q_1, w_{1Q}>;<q_2, w_{2Q}>;.....<q_i, w_{iQ}>\} \quad (1)$$

$$iqf_i = \log(\frac{n}{qf_i}) \quad (2)$$

$$w_{iQj} = tf_{iQj} * iqf_i \quad (3)$$

Given Q, we define C_{ij} as (4) which represents the common term vector of two queries Q_i and Q_j. Here, q refers to the terms that belong to both Q_i and Q_j.

$$C_{ij} = \{q : q \in Q_i \cap Q_j) \quad (4)$$

Given these concepts, we now can provide one definition of query similarity:

Definition I: A query Q_i is similar to query Q_j if $|C_{ij}|>0$, where the $|C_{ij}|$ is the number of common terms in both queries.

A basic similarity measure based on common query terms can be computed as follows:

$$Sim\_basic(Q_i, Q_j) = \frac{|C_{ij}|}{Max(|Q_i|, |Q_j|)} \quad (5)$$

where $|Q_i|$ is the number of the keywords in a query Q_i.

Taking the term weights into consideration, we can use any one of the standard similarity measures [6]. Here, we only present the cosine similarity measure since it is most frequently used in information retrieval:

$$Sim\_cosine(Q_i, Q_j) = \frac{\sum_{i=1}^{k} cw_{iQ_i} \times cw_{iQj}}{\sqrt{\sum_{i=1}^{k} cw_{iQ_i}^2} * \sqrt{\sum_{i=1}^{k} cw_{iQ_j}^2}} \quad (6)$$

where cw_{iQi} refers to the weight of i^{th} common term of C_{ij} in query Qi.

As discussed, the content-based approach is the simplest method to construct query clusters and the computational costs of using such an approach are relatively low. However its effectiveness is questionable due to the short lengths of most queries. For example the term "light" can be used in four different ways (noun, verb, adjective and adverb). In such cases, content-based query clustering may not be able to distinguish the semantic differences behind the terms due to the lack of contextual information and thus may not provide reasonable cluster results. Thus an alternative approach based on query results is considered.

3.2. The result URLs-based similarity measure

The results returned by search engines usually contain a variety of information such as the title, abstract, topic, etc. This information can be used to compare the similarity between queries. In our work, taking the cost of processing query results into consideration, we consider the query results' unique identifiers (e.g. URLs) in determining the similarity between queries as in [4, 26].

Let $U(Q_j)$ represent a set of query result URLs to query Q_j:

$$U(Q_j) = \{u_i, u_2, \ldots\ldots\ldots u_i\} \quad (7)$$

where u_i represents the i^{th} result URL for query Q_j. We then define R_{ij} as (8), which represents the common query results URL vector between Q_i and Q_j. Here u refers to the URLs that belong to both $U(Q_i)$ and $U(Q_j)$.

$$R_{ij} = \{u : u \in U(Q_i) \cap U(Q_j)\} \quad (8)$$

Next, the similarity definition based on query result URLs can be stated as:

Definition II: A query Q_i is similar to query Q_j if $|R_{ij}|>0$, where the $|R_{ij}|$ is the number of common result URLs in both queries.

The similarity measure can then be expressed as (9)

$$Sim\_result(Q_i, Q_j) = \frac{|R_{ij}|}{Max(|U(Q_i)|, |U(Q_j)|)} \quad (9)$$

where the $|U(Q_i)|$ is the number of result URLs in $U(Q_i)$. Note that this is only one possible formula for calculating similarity using result URLs. Other measures for determining the similarity can be used. For example, overlaps of result titles or overlaps of the domain names in the result URLs.

3.3. Determining query clusters

Given a set of queries $Q=\{Q_1, Q_2 \ldots Qn\}$ and a similarity measure between queries, we next construct query clusters. 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 (10). Here $Sim(Q_i, Q_j)$ refers to the similarity between Qi and Q_j which can be computed by using various similarity measures discussed previously.

$$G(Q_i) = \{Q : Sim(Q_i, Q_j) \geq threshold \ \} \quad (10)$$

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

Note there are alternative query clustering approaches besides the one used in our experiments [27]. Compared with these approaches, our method is relatively less time consuming. The advantage is that it is suitable for use in systems that need to respond and recommend queries to users in real time.

4. Query clustering experiments

Our experiments examine the effect of different similarity measures on the quality of query clusters. Here, we examine the following questions:

- To what extent do the term weights boost the performance of clustering algorithms? In spite of the success of the use of term weights in document clustering, the value of term weights remain uncertain in query clustering due to the short length of queries. However, to date, there are few studies focusing on this question. Hence, in our experiments, we compare the performance of the basic similarity measure (5) and the cosine similarity measure (6) since they are representative approaches in the literature [6, 22].
- Are there differences in cluster quality between the content-based and results-based approaches? Previous studies have focused on the comparison between feedback-based and content-based approaches [22]. Yet

in the literature presented in the previous section, it can be argued that the results-based approach play an important role in query clustering. To the best of our knowledge, there is little work done on comparing as well as quantifying the differences between the content-based and results-based approaches.

4.1. Data set and data preprocessing

We collected six-month query logs (around two million query sessions) from the digital library at Nanyang Technological University (Singapore). The query logs are in text format and contain information such as the time when the user issued the query, the query terms submitted to the search engine and the number of returned results by the search engine in response to the query terms.

We preprocessed the query logs according to the following steps:

- In order to reduce the size of the raw data, only relevant data was extracted. This included the query terms and the corresponding number of hits.
- Due to the large number of queries contained in the query logs, sampling was carried out. Previous studies indicate that the query sample sizes will impact the final experiment results [4]. Thus 35000 queries from the query log were selected for our evaluation, consistent with previous studies' sample sizes of between several hundred [7, 25] to tens of thousands of queries [4, 22]. Further, all identical queries were removed. This reduced the sample to 17000 queries. Note that there were more than 50% of the queries in the original query set that have been repeated over time. This phenomenon indicates, to a certain extent, user's interests tend to overlap, hence reinforcing the usefulness of utilizing previously issued queries to facilitate information seeking.
- Since the search engine offers advanced search options, some of the queries have prefixes representing these options. For example, "ti" indicates that this search is within the title field. These prefixes were removed from queries and only the query terms were retained.
- The queries that contain misspelled terms were removed since no documents were retrieved, posing difficulties for the results-based similarity measure. After this step, there were around 16000 distinct queries left for our experiments.
- Stop words were removed from the queries in order to get better clustering results when using content-based similarity measures since these terms (such as "the", "a", "an") do not convey useful meanings.

Within the 16000 query samples, 23% of the queries contained one keyword, 36% of the queries contained two keywords, and 18% of the queries contained three keywords. Further, approximately, 77% of the queries contained no more than three keywords. The average length of all the query samples was 2.73. This observation is similar to previous studies [23]. The 16000 query samples contained 37752 individual terms. It is interesting to observe that there were 9503 distinct terms within the 16000 query samples. Therefore, each distinct term

appeared 3.97 times on average. This observation shows that people tend to use similar keywords to express their information needs. Table 1 shows some examples in the final query sample.

cards game	fabrication of CMOS	mobile phone works
communications between people	handbook chemical engineering	NiTi matrix composites
desalination plant costs	intelligence and gene	packaging machinery
device and material characterization	Julius Lester	process of water treatment

Table 1. Examples of queries

4.2. Methodology

We calculated the similarity between queries using the following similarity measures:

- Basic similarity (sim_basic) ── function (5)
- Content-based similarity (sim_cosine) ── function (6)
- Results-based similarity (sim_result) ──function (9)

First, all queries were split into separate terms. For sim_basic, each query length was computed. Next, the number of common terms between two queries was derived by calculating the intersection of the two queries. Finally, the sim_basic measure was computed. For sim_cosine, the weight of all terms within a single query was computed using function (2). By using the intersection of the two queries generated in the previous step as well as the weight of each term, sim_cosine was computed by using function (6). For sim_result, we posted each query to a reference search engine (Google) and retrieved the corresponding result URLs. By design, search engines rank highly relevant results higher, and therefore, we only considered the top 10 result URLs returned to each query. This method is similar to those used in [4, 26]. The result URLs were then used to compute the similarity between queries according to function (9).

Recall that two queries are in one cluster whenever their similarity is above a certain threshold (10). Threshold is the minimum value, obtained from a given similarity measure, that determines whether two queries should be clustered into to the same group. Therefore, different thresholds will lead to different query clusters. In all similarity measures, thresholds were set to 0.25, 0.5, 0.7 and 0.9.

4.3. Measures of cluster quality

In our experiments, the quality of query clusters using different similarity measures was examined. Here, quality is measured by average cluster size, coverage, precision and recall.

After obtaining the clusters based on the different similarity measures, we first observed the average cluster size. This information sheds light on the ability of the different measures to provide recommended queries to a given query. In other words, this value reflects the variety of recommended queries to a user.

Next, coverage, precision and recall were calculated. Coverage is the ability of the different similarity measures to find similar queries for a given query. It is the percentage of queries for which the similarity measure is able to provide a cluster. This value will indicate the probability that the user can obtain recommended queries for his/her issued query.

Precision and recall were used to assess the accuracy of the query clusters generated by the different similarity measures. Precision refers to the ratio of the number of similar queries to the total number of queries in a cluster. For precision, we randomly selected 100 clusters and checked each query in the cluster manually as done in [22]. Since the actual information needs represented by the queries are not known, the similarity between queries within a cluster was judged by a human evaluator who took into account the query terms as well as the result URLs. The average precision was then computed for the 100 selected clusters.

Recall refers to the ratio of the number of similar queries to the total number of all similar queries across the query set (those in the current cluster and others). However its calculation posed a problem as it was difficult to compute directly because no standard clusters were available in the query set. Therefore, an alternative measure of recall was used. Here, recall is defined to be the ratio of the number of correctly clustered queries within the 100 selected clusters to the maximum number of the correctly clustered queries across the test collection, similar to [22]. The number of correctly clustered queries within the 100 selected clusters equals to the total number of queries in the 100 selected query clusters multiplied by the average precision. Note that the number of queries in the 100 selected query clusters can be computed by the average cluster size multiplied by 100. In our work, the maximum number of the correctly clustered queries was 1948, which was obtained by sim_basic with the threshold of 0.25.

Analysis of variance procedures (ANOVA) were also conducted to reveal whether the different thresholds and similarity measures significantly affected the quality of query clusters in terms of average cluster size, precision and recall. Since the values for coverage are categorical, the Chi-squared test was used to measure the effect of thresholds and similarity measures on coverage.

5. Experimental findings

In this section, we describe the results of our experiments and compare the similarity measures against the different quality criteria described previously. Further, we discuss the underlying reasons for these results.

5.1. Results

By varying the similarity thresholds, we obtained different average cluster sizes as shown in Figure 1. Along with the change of threshold from 0.25 to 0.9, the average cluster size of sim_basic decreases from 50.27 to 2.11, sim_cosine decreases from 43.65 to 8.06 and sim_result decreases from 2.63 to 2.21. It can be seen from the figure that when using sim_basic and sim_cosine to cluster queries, the average cluster size is larger than using sim_result. This indicates that for a query cluster, the content-based approach (both sim_basic and sim_cosine) can find a larger number of queries for a given query than the results-based approach (sim_result). Stated differently, the content-based approach can provide a greater variety of queries to a user given his/her submitted query. It is interesting to observe that sim_basic outperforms sim_cosine when the threshold is less than 0.6 while sim_cosine performs better when the threshold is larger than 0.6. The reason behind this phenomenon may be that as the number of common terms between two queries increase, their term weights play a more important role in finding related queries.

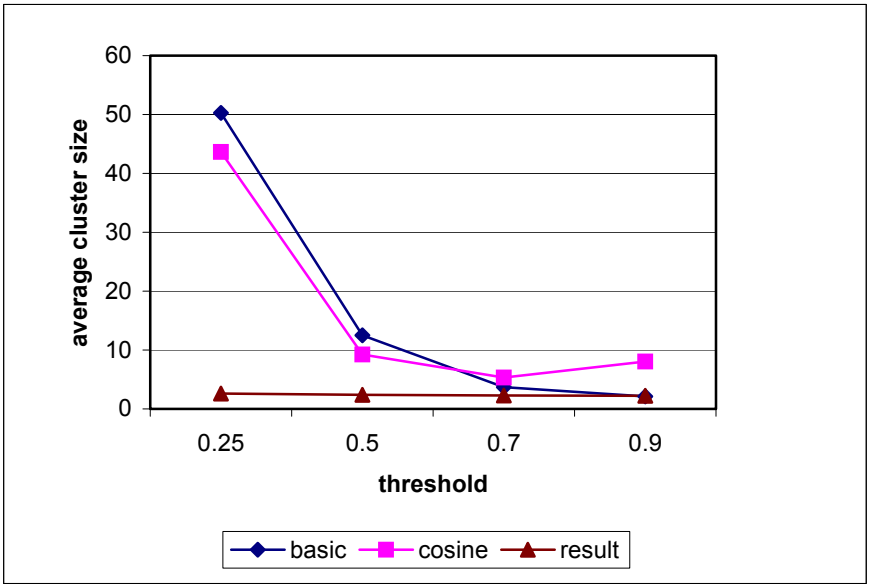

Fig 1. Average cluster sizes

A 4 X 3 (thresholds X similarity measures) ANOVA yielded a statistically significant interaction effect on average cluster size, $F(6,11) = 306.98$, $p < .001$. There also existed significant effects for thresholds, $F(3,11) = 4205.66$, $p < .001$, and for similarity measures $F(2,11) = 1353.50$, $p < .001$. This indicates that the difference in

the average cluster sizes is significant across the cells defined by the combination of the two factor levels: thresholds and similarity measures.

For coverage, sim_basic decreases from 80.45% to 3.71%, sim_cosine decreases from 82.74% to 18.02% and sim_result decreases from 22.03% to 6.99%, with the change of threshold from 0.25 to 0.9 respectively (see Figure 2). This shows that sim_cosine and sim_basic rank higher in coverage, demonstrating that the content-based approach has a better ability to find similar queries from a given query than the results-based approach. The fact, as discovered during processing of the query logs and as discussed previously, that users tend to use similar terms to express their information needs might account for the high performance of the content-based approaches in term of coverage. On the other hand, the number of distinct URLs is often large. This might explain the poorer performance of sim_result in terms of coverage since many similar queries cannot be grouped together due to a lack of common result URLs [26]. Further, sim_cosine performs better than sim_basic through all threshold levels indicating that term weights can improve the ability to find similar queries from a given query in spite of the short length of queries.

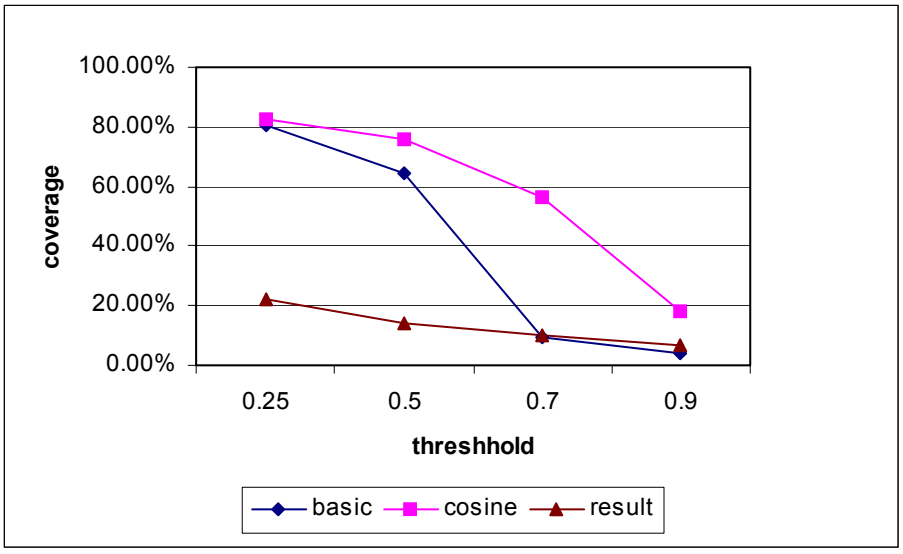

Fig 2. Coverage

The Chi-squared test confirmed that for each threshold, the differences in coverage across various similarity measures were significant: X^2 (2, N=48000) = 15886.61, $p < .001$; X^2 (2, N=48000) = 27056.64, $p < .001$; X^2 (2, N=48000) = 12124.40, $p < .001$; X^2 (2, N=48000) = 2069.60, $p < .001$ with thresholds of 0.25, 0.5, 0.7 and 0.9 respectively.

Figure 3 indicates that the results-based approach is better able to cluster similar queries correctly than the other measures. In terms of precision, sim_result increases from 93.33% to 100%, along with the change of similarity threshold from 0.25 to 0.9. This indicates that almost all of the queries in the cluster were considered similar. When the threshold equals 0.9, the precision of sim_result reaches the peak, 100%, which indicates that there are

no "irrelevant queries" in the clusters. This time, the content-based approaches suffer from poorer performance in terms of precision. The precision of sim_basic increases from 38.74% to 99.98% and sim_cosine increases from 35.46% to 96.56%. The precision of both sim_cosine and sim_basic are consistently below that of sim_result. The precision of the content-based approaches is lower because of the short length of queries and the lack of the contextual information in which queries are used. On other hand, for sim_result, the reference search engine (Google) tends to return the same URLs to semantically related queries [4, 26], which might account for the good performance of the results-based approach in terms of precision.

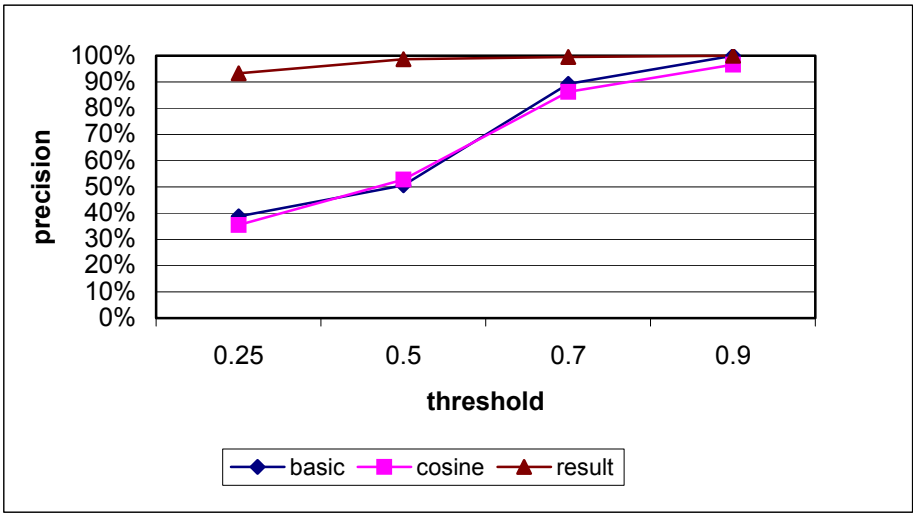

Fig 3. Precision

A 4 X 3 (thresholds X similarity measures) ANOVA yielded a statistically significant interaction effect on precision, F (6,11) = 42.41, p < .001. There also existed main effects for thresholds, F (3,11) = 211.45, p < .001, and for similarity measures F (2,11)=192.52, p < .001. This indicates that the precision is significantly different across the cells defined by the combination of the two factor levels: thresholds and similarity measures.

For recall, sim_basic has the best performance at 100% when the threshold equals 0.25, indicating that all similar queries were contained in the query clusters. It is interesting to observe that sim_basic outperforms sim_cosine when the threshold is less than 0.6 while sim_cosine performs better when the threshold is larger than 0.6. The reason behind this phenomenon is that since the recall calculation includes average cluster size (refer to the definition of recall in Section 4.3), therefore, recall changes in accordance with average cluster size (see Figure 1). Further both sim_basic and sim_cosine outperform sim_result in terms of recall. The low average cluster size of sim_result might account for this. Note that although the recall used in this experiment is not the same as the traditional definition used in information retrieval research, it does provide useful information to indicate the accuracy of clusters generated by the different similarity measures [22]. That is, the modified recall measure reflects the ability to uncover clusters of similar queries generated by the different similarity measures on the sample set of queries used in the experiments.

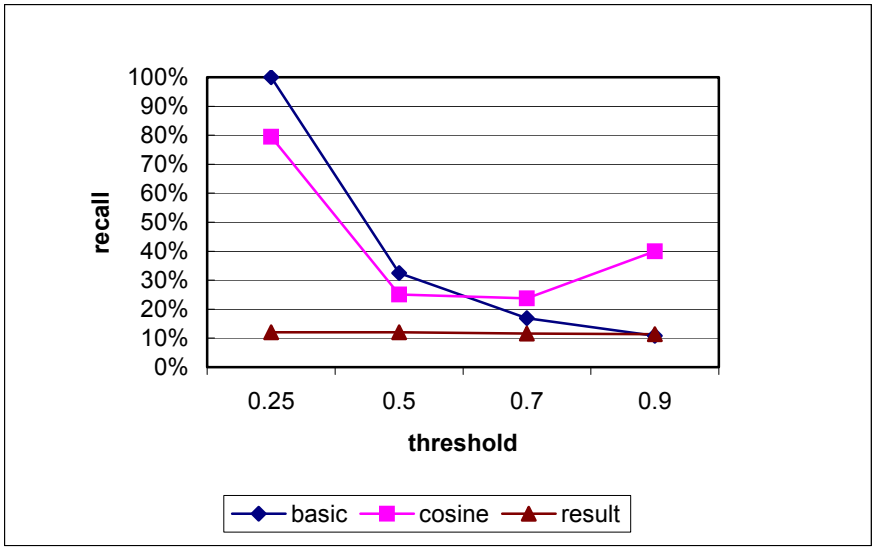

Fig 4. Recall

A 4 X 3 (thresholds X similarity measures) ANOVA yielded a statistically significant interaction effect on precision, F (6,11) = 206.89, $p < .001$. There also existed main effects for thresholds, F (3,11) = 1057.12, $p < .001$, and for similarity measures F (2,11) = 423.14, $p < .001$. This indicates that the differences in recall are significant across the cells defined by the combination of the two factor levels: thresholds and similarity measures.

5.2. Discussion

Our experiments show that there is no single best similarity measure for clustering queries, since for each criterion used to assess cluster quality, different measures outperform each other. Table 2 summarizes the comparison between the content-based and results-based approaches. Here, the average value across all thresholds in terms of the different performance measures was used to generate this table. The approach whose average value is larger is regarded as better. For example, users will obtain larger average cluster sizes using the content-based approach while the precision of the recommended queries will be poorer. On the other hand, the results-based approach improves average precision but suffers from poorer coverage and recall. This observation offers opportunities to enhance the performance of query clustering algorithms by using both query terms and their result URLs since the strengths of each similarity measure might counteract against the drawbacks of the other.

	Average cluster size	Coverage	Precision	Recall
Better	Content-based approach	Content-based approach	Results-based approach	Content-based approach
Worse	Results-based approach	Results-based approach	Content-based approach	Results-based approach

Table 2. Comparison between the content-based and results-based approach

Further, when compared with sim_basic, our experiments show that sim_cosine improves the coverage of query clusters without sacrificing the other aspects of cluster quality. Table 3 compares the different approaches in greater detail, taking the impact of the different thresholds into consideration. The thresholds were categorized into two groups: low threshold (0.25 and 0.5) and high threshold (0.7 and 0.9). Letters within each cell indicate the performance of the similarity measure against a cluster quality criterion. For example, for low threshold and average cluster size as the cluster quality criterion, sim_basic ranks first followed by sim_cosine and sim_result, hence B, C, R. Note that for precision, B(C) means that sim_basic and sim_cosine have the same performance.

B─sim_basic, C─sim_cosine, R─sim_result

	Average cluster size	Coverage	Precision	Recall
Low threshold (0.25, 0.5)	B, C, R	C, B, R	R, B(C)	B, C, R
High threshold (0.7, 0.9)	C, B, R	C, R, B	R, B(C)	C, B, R

Table 3. Comparison of different similarity measures categorized by threshold

6. Conclusion

In this paper, we compared the impact of different query similarity measures on the quality of query clusters. Similarity is fundamental to the construction of such clusters and measures of similarity between queries are therefore essential in the query clustering process. Because of the little work done in this field, we sought to carry out a rigorous analysis on the performance of different similarity measures. Our experiments show that by using the content-based and results-based approaches alone, each method possesses drawbacks that will affect the quality of query clusters. From the results, it can be deduced that the precision of the content-based approach is

low due to the short length of queries and the lack of the contextual information in which queries are used, while the results-based approach performs well in terms of precision since the reference search engine used tends to return the same URLs to semantically related queries. On the other hand, the results-based approach suffers from poor performance in terms of coverage since many similar queries cannot be grouped together due to a lack of common result URLs, while the content-based approach performs better because users tend to express their information needs using similar terms. Taken together, this suggests that the two approaches may complement each other, each offering advantages that may compensate for the weaknesses of the other to improve upon the overall quality of the query clusters. Further, sim_cosine and sim_basic generate similar results in almost all aspects of cluster quality except coverage, in which sim_cosine outperforms sim_basic. This suggests that term weights can contribute to the quality of 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. The results reported here can be used to develop new systems or further refine existing systems that determine and cluster similar queries in query logs, and augment the information seeking process by recommending related queries to users. As discussed previously, these systems can help information seekers, especially novices, to express their information needs.

In addition to the initial experiments performed in this research, experiments involving hybrid approaches that exploit both query terms as well as result URLs are also planned. Building upon the content-based and results-based approaches, the hybrid approach might generate better quality query clusters than using each approach individually. Further, alternative definitions of similarity between queries will also be investigated. For example, the result URLs can be replaced by the domain names of the URLs to improve the coverage of the results-based query clustering approach. In addition, experiments using other clustering algorithms such as DBSCAN [28] might also be conducted to assess cluster quality. Since DBSCAN is a density-based clustering algorithm, indirectly related queries to the current query may be found, hence improving average cluster size and coverage. Finally, word relationships such as hypernyms and synonyms can be used to replace query terms before computing the similarity between queries to increase the coverage as well as average cluster size.

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

[1] I.M. Lokman & W.H. Stephanie, Information–seeking behavior and use of social science faculty studying stateless nations: a case study, *Journal of Library and Information Science Research* 23(1), (2001), 5-25.

[2] G.N. Marchionini, *Information seeking in electronic environments* (Cambridge University Press, Cambridge, 1995).

[3] R. Taylor, Question-negotiation and information seeking in libraries, *College and Research Libraries* 29(3), (1968), 178-194.

[4] N.S. Glance, Community search assistant, *Proceedings of the 6th ACM International Conference on Intelligent User Interfaces* (Santa Fe, January 2001), 91-96.

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

[6] G. Salton & M.J. McGill, *Introduction to modern information retrieval* (McGraw-Hill New York, NY, 1983).

[7] V.V. Raghavan, & H. Sever, On the reuse of past optimal queries, *Proceedings of the 18th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (Seattle, July 1995), 344-350.

[8] T.D. Wilson, Human information behaviour, *Informing Science* 3(2), (2000), 49-55.

[9] B. Dervin & P. Dewdney, Neutral questioning: a new approach to the reference interview, *Reference Quarterly* 25(4), (1986), 506-513.

[10] D. Ellis, A comparison of the information seeking patterns of researchers in the physical and social sciences, *Journal of Documentation* 49(4), (1993), 356-369.

[11] H. Lieberman, An agent for web browsing, *Proceedings of the International Joint Conference on Artificial Intelligence* (Montreal, August 1995), 924-929.

[12] G.D.J.H. Revera, J. Courtiat & T. Villemur, A design framework for collaborative browsing, *Proceedings of the 10th IEEE International Workshop on Enabling Technologies: Infrastructure for Collaborative Enterprises* (Cambridge, June 2001), 362-374.

[13] WebEx home page. Available at www.webex.com (accessed 15 May 2003).

[14] I.G. Chun & I.S. Hong, The implementation of knowledge-based recommender system for electronic commerce using Java expert system library, *Proceedings of the IEEE International Symposium on Industrial Electronics* (Pusan, June 2001), 1766-1770.

[15] D. Goldberg, D. Nichols, B.M. Oki & D. Terry, Using collaborative filtering to weave an information tapestry, *Communications of the ACM* 35(12), (1992), 61-70.

[16] P. Resnick, N. Iacovou, S. Mitesh, P. Bergstron & J. Riedl, GroupLens: an open architecture for collaborative filtering of Netnews, *Proceedings of the 1994 ACM Conference on Computer Supported Cooperative Work* (Chapel Hill, October 1994), 175-186.

[17] L. Teveen, W. Hill, B. Amento, M. David & J. Creter, PHOAKS: a system for sharing recommendations, *Communications of the ACM* 40(3), (1997), 59-62.

[18] J. Pomerantz & R.D. Lankes, Integrating expertise into the NSDL: putting a human face on the digital library, *Proceedings of the 2nd Joint Conference on Digital Libraries* (Portland, July 2002), 405.

[19] B. Sloan, *Service perspectives for the digital library remote reference services* (1997). Available at: www.lis.uiuc.edu/~b-sloan/e-ref.html (accessed 25 December 2002).

[20] E. Anderson, J. Boyer & K. Ciccone, Remote reference services at the North Carolina State University Libraries, *Proceedings of the 2nd Digital Reference Conference* (Seattle, October 2000), 20-28.

[21] C.J. Crouch, D.B. Crouch, & K.R. Kareddy, The automatic generation of extended queries, *Proceedings of the 13th Annual International ACM SIGIR Conference* (Brussels, September 1990), 269-283.

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

[23] C. Silverstein, M. Henzinger, H. Marais & M. Moricz, Analysis of a very large Altavista query log, *DEC SRC Technical Note* 1998-14, (1998).

[24] S.L. Chuang & L.F. Chien, Towards automatic generation of query taxonomy: a hierarchical query clustering approach, *Proceedings of the IEEE 2002 International Conference on Data Mining* (Maebashi, December 2002), 75-82.

[25] L. Fitzpatrick & M. Dent, Automatic feedback using past queries: social searching? *Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (Philadelphia, July 1997), 306-313.

[26] R.Z. Osmar & S. Alexander, Finding similar queries to satisfy searches based on query traces, *Workshops of the 8th International Conference on Object-Oriented Information Systems* (Montpellier, September 2002), 207-216.

[27] A.K. Jain, M.N. Murty & P.J. Flynn, Data clustering: a review, *ACM Computing Surveys* 31(3), (1999), 264-323.

[28] M. Ester, H. Kriegel, J. Sander & X. Xu, A density-based algorithm for discovering clusters in large spatial databases with noise, *Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining* (Portland, August 1996), 226-231.