The Effect of Similarity Measures on The Quality of Query Clusters

Lin Fu, Dion Hoe-Lian Goh, Schubert Shou-Boon Foo, Jin-Cheon Na
Division of Information Studies
School of Communication and Information
Nanyang Technological University, Singapore 637718

Abstract

Query clustering is a process to group similar queries automatically into different categories. This task is important to discover the common interests of online information seekers and to exploit the experience of previous users for the others, which are harnessed to facilitate collaborative querying that can help users in digital libraries and other information systems better meet their information needs. In such cases, the kernel step is to identify the similarity measure between queries. In this paper, we examine the effectiveness of different similarity identification methods. A set of experiments has been carried out to study the impact of different similarity measures on the final query clustering performance.

1. Introduction

With the increasing proliferation of Internet, people have now come to depend more on the Web or digital libraries (DLs) to search for information. Yet the performance of the 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 the material in search of relevant content.

The study of information seeking behavior has revealed that interaction and collaboration with other people is an important part in the process of information seeking and use [7][8][17]. Given this idea, collaborative search aims to support collaboration among people when they search information on the Web or in DLs [5]. Work in collaborative search falls into several major categories including collaborative browsing, collaborative filtering and collaborative querying [14]. 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 [14]. Query mining is one of the common techniques used to support collaborative querying. It allows users to make use of 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 is given below.

Consider a user A that is interested in the XML parser for Java programming, and she wants to look for articles and useful web resources relevant to this field. Due to her limited domain knowledge, she enters “XML parser” as the query to her preferred search engine and gets lists 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 know that good search results can be obtained by using “JDOM” as the query. Note that B’s search history is usually stored in the 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 query logs, clustering similar queries and then recommending them to users, there is an opportunity for the first user to take advantage of previous queries that someone else had entered 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 area suggests a similarity calculation between queries according to query terms (content-based approach) [13]. The queries will be grouped into one 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) [5][10]. In such 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 performances based on different query similarity calculation approaches. Therefore, the effect of different query similarity identification approaches on the quality of query clusters has not been studied to date. In this paper, a comprehensive evaluation on different query similarity calculation methods is reported.

This work will benefit information retrieval systems and DLs in better meeting the information needs of users through collaborative querying. Specifically, this work reveals the drawbacks and advantages of different query similarity calculation approaches and shed light on improving the performance of the algorithms adopted by query recommending 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 identification approach adopted in the research and the algorithm to cluster queries. Then, we describe the design of evaluation experiments. Further, we report experimental results that assesses the effectiveness of different approaches. 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 behavior 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 Behavior

Information seeking is a broad term encompassing the ways individuals articulate their information needs, seek, evaluate, select and use information (Lokman & Stephanie, 2001). In other words, information seeking behavior 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 (Wilson, 2000). 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 (1968) 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.

Similarly, Dervin and Dewdney’s (1986) 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 behavior. 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 behavior is communicating behavior, 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, Elise’s (1993) research resulted in a pattern of information-seeking behavior 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. Elise 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 which seeks to support cooperation among people when they search information on line. It can be divided into three types according to the ways that users search for information: collaborative browsing, collaborative querying and collaborative filtering [14]. 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 [6]. 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” [6], a system for co-located collaborative browsing using user interests, and “WebEx” [19], a meeting system that allows distributed users to browse a Web pages.

Collaborative filtering is a technique for recommending items to a user based on similarities between the past behavior of the user and that of likeminded people [1]. 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 [4]. Examples of collaborative filtering applications include Tapestry [4], a system that can filter information according to other users’ annotations; GroupLens [12], a recommender system using user ratings of documents read; and PHOAKS [18], 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 [9]. 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-half mile away in a College of Medicine computer lab [16].

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 [21].
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 [5]. The system can then either recommend the similar queries to users (query recommending systems) [5] or use them as expansion term candidates to the original query to augment the quality of the search results (automatic query expansion systems) [10][24]. Such an approach overcomes the limitation of human involvement and network overloading inherent in online live reference service. 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

2.3.1 Content-based approaches

Traditional information retrieval research suggests an approach to query clustering by comparing query term vectors (content-based approach). 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 functions incorporating the consideration of term weights are available including cosine-similarity, Jaccard-similarity, and Dice-similarity [13]. Using these functions have provided good results in document clustering due to the large number of terms contained in documents. Such kind of method is simple and straightforward 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 [20]. 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 [15]. 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 [10].

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 [20]. 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 [20]. Further, the performance of such methods will be affected greatly by the lack of common documents clicked by users [22]. In other words, if users click different documents for
the identical or similar queries, such methods will not generate effective query clusters.

2.3.3 Results-based approaches

Raghavan and Sever [10] determine similarity between queries by calculating the overlap in documents returned by the queries. This is done by converting result documents into term frequency vectors. Then the similarity between two queries was decided by comparing the query result vectors rather than treating the queries as term-vectors. Fitzpatrick and Dent [3] further develop 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. The weights used in their experiment are empirically derived probabilities of different result list ranges to contain relevant documents. Using the corresponding query results to cluster queries is useful in boosting the performance of query clustering in terms of precision and recall [3][10]. However this method is time consuming to perform and is not suitable for online search systems [3]. Glance [5] thus uses the overlap of result URLs as the similarity measure instead of the document content. Queries were 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 Calculations

This section provides definitions of different query similarity identification approaches used in our evaluation experiments. Further, the definition of how we construct query clusters based on different query similarity measures is presented.

3.1 Content-based Similarity Approach

We borrow concepts from information retrieval [13] and define a set of queries as \( D = \{Q_1, Q_2, \ldots, Q_i, Q_j, \ldots, Q_n\} \). 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 \( w_{iQ_j} \) 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 (Stated alternatively, they are important to express the information needs of a query and valuable to cluster queries). High query frequency, on the other hand, indicates that a term is too general to be useful as descriptor (In other words, they will not convey useful information for query clustering). Next, the inverse query frequency, \( iqf_i \), is expressed as (3), in which \( n \) represents the total number of queries in the query collection. We then compute \( w_{iQj} \) based on (2):

\[
Q_j = \{<q_1, w_{Qj}>, <q_2, w_{Qj}>, \ldots, <q_i, w_{Qj}>\} \tag{1}
\]

\[
w_{iQj} = tf_{iQj} * iqf_i \tag{2}
\]
Given D, 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 $Q_j$.

$$C_{ij} = \{q : q \in Q_i \cap Q_j\}$$

(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 query terms can be computed as follows:

$$\text{Sim}_{\text{basic}}(Q_i, Q_j) = \frac{|C_{ij}|}{\text{Max}(|Q_i|, |Q_j|)}$$

(5)

where $N(Q_i)$ 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 [13]. Here, we only present the cosine-similarity measure since it is most frequently used in information retrieval:

$$\text{Sim}_{\text{cosine}}(Q_i, Q_j) = \frac{\sum_{i=1}^{k} c_{wiQ_i} \times c_{wiQ_j}}{\sqrt{\sum_{i=1}^{k} c_{wiQ_i}^2} \times \sqrt{\sum_{i=1}^{k} c_{wiQ_j}^2}}$$

(6)

where $c_{wiQ_i}$ refers to the weight of $i^{th}$ common term of $C_{ij}$ in query $Q_i$.

As discussed, the content-based approach is the simplest method to construct query clusters and the costs of using such an approach is 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 cannot distinguish the semantic differences behind the terms due to the lack of contextual information and thus cannot provide reasonable cluster results. Thus an alternative approach based on query results is considered.

**3.2 Result URLs-based Similarity Approach**

The results returned by search engines usually contain a variety of information such as the title, the abstract, the category, etc. This information can be used to compare the similarity between queries. In our work, taking the cost of performing time into
consideration, we consider the query results’ unique identifiers (e.g. URLs) in determining the similarity between queries [5][23].
Let \( U(Q_j) \) be represented as set of query result URLs to query \( Q_j \):

\[
U(Q_j) = \{u_i, u_2, \ldots, u_n\}
\]

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 \( U(Q_j) \).

\[
R_{ij} = \{u : u \in U(Q_i) \cap U(Q_j)\}
\]

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)

\[
\text{Sim}_{\text{result}}(Q_i, Q_j) = \frac{|R_{ij}|}{\max(|U(Q_i)|, |U(Q_j)|)}
\]

where the \( |U(Q_i)| \) is the number of result URLs in \( U(Q_i) \). Note that this is only one possible formula of 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 \( D = \{Q_1, Q_2, \ldots, Q_n\} \) 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 (11). Here \( \text{Sim}(Q_i, Q_j) \) refers to the similarity between \( Q_i \) and \( Q_j \) which can be computed by using various similarity functions discussed previously.

\[
G(Q_j) = \{Q : \text{Sim}(Q_i, Q_j) \geq \text{threshold}\}
\]

where \( 1 < j < n; n \) is the total query number. Note there are alternative query clustering approaches besides the one used in our experiments, for example, Hierarchical Agglomerative Clustering (HAC) algorithms [25]. Comparing with other approaches, our method is relatively less time consuming; thus, the query clusters can be easily constructed.

4. Query Clustering Experiments

In our experiments, we want to examine the following questions.

- To what extent 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 basic similarity measure and the cosine similarity measure since they are representative approaches in literature [20][13].

• Are there differences in cluster quality between the content-based approach and results-based approach? Previous studies have focused on the comparison between feedback-based approach and content-based approach [20]. Yet according to the literature presented in previous section, it is obvious that results-based approach plays an important role in query clustering. To the best of our knowledge, there is little work on comparing as well as quantifying the differences between the content-based approach and results-based approach. It is will be interesting to conduct such an experiment to reveal the strength and weakness behind these two approaches.

4.1 Data Set & Data Preprocessing

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

We preprocessed the raw query logs according to the following steps:

• In order to reduce the size of the raw data, all relevant data was extracted. This includes the query terms and the corresponding records number.

• Due to the large amount of queries contained in the query log, sampling was carried out. Previous studies indicate that the query sample sizes will impact the final experiment results [5]. Thus 35000 queries from the query log were selected for our evaluation since previous studies’ sample sizes vary from several hundred [3][10] to tens of thousand queries [5][20]. Further, all identical queries were removed so that the queries were distinct from each other. Therefore, the size of queries was decreased to 17000. Note that there are more than 50% queries in the original query set, which have been repeated over time. This phenomenon indicates, to a certain extent, user’s interests tend to be overlapped and reinforce the usefulness of utilizing previous issued queries to facilitate a successful information seeking.

• Since the search engine offers advanced search options by which users can choose a specific domain to search for information, some of the queries have a prefix, which indicates the specific domain to search. Such kind of options is embedded in the query terms. For example, “ti” indicates that this search is within the title field. Thus, these prefixes were removed from queries and only the real query terms were remained.

• The queries that contain misspelling terms were removed since they do not make any sense and no documents were retrieved. After this step, there were around 16000 distinct queries left for our experiment.

• 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 [15]. 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 appears 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.

**Table 1. Examples of Queries**

<table>
<thead>
<tr>
<th>cards game</th>
<th>fabrication of CMOS</th>
<th>mobile phone works</th>
</tr>
</thead>
<tbody>
<tr>
<td>communications between people</td>
<td>handbook chemical engineering</td>
<td>NiTi matrix composites</td>
</tr>
<tr>
<td>desalination plant costs</td>
<td>intelligence and gene</td>
<td>packaging machinery</td>
</tr>
<tr>
<td>device and material characterization</td>
<td>Julius Lester</td>
<td>process of water treatment</td>
</tr>
</tbody>
</table>

### 4.2. Methodology

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

- Basic similarity ($\text{sim}_{\text{basic}}$) -- function (5)
- Content-based similarity ($\text{sim}_{\text{cosine}}$) -- function (6)
- Results-based similarity ($\text{sim}_{\text{result}}$) -- function (9)

First, all queries were split into separate terms. For $\text{sim}_{\text{basic}}$, each query length was computed. Next, the number of common terms between two queries was computed by calculating the intersection of two queries. Finally, the $\text{sim}_{\text{basic}}$ function was calculated. For $\text{sim}_{\text{cosine}}$, the weight of all terms within a single query was computed using function (2). By using the intersection of two queries generated in the previous step as well as the weight of each term, $\text{sim}_{\text{cosine}}$ was computed by using function (6). For $\text{sim}_{\text{result}}$, 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 [5][23]. The result URLs were then be 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. Threshold is the baseline to determine whether two queries should be clustered into to the same group. Therefore, different thresholds will lead to different query clusters. In all approaches, similarity thresholds (10) were set to 0.25, 0.5, 0.7 and 0.9 respectively in order to study the impact of various thresholds on the final performance of the clustering algorithms.
4.3. Performance Measures

In our experiments, the quality of query clusters using different similarity calculation approaches was examined (please refer to Introduction Section). After obtaining the clusters based on the different similarity measures, we first observed the average cluster size and the range of the cluster sizes. This information sheds light on the ability of the different measures to provide recommended queries on a given query. In other words, they can reflect the variety of the 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 function 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 different similarity functions. First, precision referred 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 [20]. 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 by taking into account the query terms as well as 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 it posed a problem as it was difficult to calculate directly because no standard clusters were available in the query set. Therefore, an alternative measure to reflect recall was used. Recall was 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 [20]. The number of correctly clustered queries within the 100 selected clusters equals to the query numbers of 100 selected query clusters times average precision. The query numbers of 100 selected query clusters can be computed by average cluster size times 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 thresholds and different similarity calculation approaches affected the query clusters in terms of average cluster size, precision and recall. Since the values for coverage are categorical, Chi-Square was used to measure the effect of thresholds and different similarity calculation approaches on coverage.
5. Experimental Findings

5.1 Results

By varying the similarity thresholds we obtained different average cluster sizes (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 results that when using sim_basic and sim_cosine to cluster queries, the average cluster size is bigger 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 other approaches. 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 bigger than 0.6. The reason behind this phenomenon may be that the more common terms between two queries, the more important role the weight of terms plays in finding related queries.

A 4 X 3 (4 Thresholds X 3 Similarity approaches) ANOVA yielded a statistically significant interaction effect on average cluster size, \( F(6,11) = 306.98, p < .001 \). This indicates that the variance of the average cluster sizes is significant across the cells defined by the combination of factor levels: thresholds and similarity approaches. There also existed significant effects for thresholds, \( F(3,11) = 4205.66, p < .001 \), and for similarity approaches \( F(2,11) = 1353.50, p < .001 \).

Further, 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 (see Figure 2). The results show that
sim_cosine and sim_basic ranks higher in coverage, demonstrating that the content-based approach has a better ability to find similar queries from a given query than results-based approach. In other words, users have a higher likelihood to obtain a recommendation to a given query than using results-based approach. The fact, as discussed previously, that the users tend to use similar terms to express their information need might account for the high performance of content-based approach in term of coverage. On the other hand, the number of distinct URLs is often huge. This might explain the low performance of sim_result in terms of coverage since many similar queries cannot be grouped together due to a lack of common result URLs [23]. Further, sim_cosine performs better than sim_basic through all the thresholds which indicates that the weight of terms can improve the ability to find similar queries from a given query in spite of the short length of queries.

The Chi-Square test indicates that for each individual threshold, the differences across various approaches are significant. For threshold of 0.25, $X^2 (2, N=48000) = 15886.61, p < .001$, for threshold of 0.5, $X^2 (2, N=48000) = 27056.64, p < .001$, for threshold of 0.7, $X^2 (2, N=48000) = 12124.40, p < .001$, for threshold of 0.9, $X^2 (2, N=48000) = 2069.60, p < .001$. This means the thresholds and different similarity identification approaches will affect the coverage significantly.

Figure 2. Coverage

Figure 3 indicates that the results-based approach is better able to cluster similar queries correctly than the other approaches. 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) performs best, indicating 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 method suffers from poorer performance in terms of precision. The precisions of sim_basic (increases from 38.74% to 99.98%) and sim_cosine (increases from 35.46% to 96.56%) generate almost same results, both of which are below that of sim_result. The precision of content-based approach is lower because of the short length of queries and the lack of the contextual information in which queries are used. On other hand, Google tends to return the same URLs to
semantically related queries [5][23], which might account for the good performance of results-based method in terms of precision.

A 4 X 3 (4 Thresholds X 3 Similarity approaches) ANOVA yielded a statistically significant interaction effect on precision, \( F (6,11) = 42.41, p < .001 \), indicating that the variance of the precision is significant across the cells defined by the combination of factor levels: thresholds and similarity approaches. There also existed main effects for thresholds, \( F (3,11) = 211.45, p < .001 \), and for similarity approaches \( F (2,11)=192.52, p < .001 \). This means each of the two factors can affect the precision significantly.

![Figure 3. Precision](image)

For recall, sim_basic has the best performance at 100% when the threshold equals 0.25, indicating that all similar queries were contained in 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 is changed 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 with the traditional definition used in information retrieval research, it does provide useful information to indicate the accuracy of clusters generated by the different similarity functions [20]. That is, the modified recall measure reflects the ability to uncover clusters of similar queries generated by different similarity functions on the sample set queries used in the experiments.

A 4 X 3 (4 Thresholds X 3 Similarity approaches) ANOVA yielded a statistically significant interaction effect on precision, \( F (6,11) = 206.89, p < .001 \), indicating that the variance of the recall is significant across the cells defined by the combination of factor levels: thresholds and similarity approaches. There also existed main effects for thresholds, \( F (3,11) = 1057.12, p < .001 \), and for similarity approaches \( F (2,11) = 423.14, p < .001 \). This means each of the two factors can affect the recall significantly.
5.2 Discussion

In summary, our experiments show that it is difficult to find a best approach by using individual similarity approach alone since for each metric in our experiments, we get different approaches which outperform others. Table 2 summarizes the comparison of content-based and results-based approaches. Here, the average value across all thresholds in terms of different performance measures was used to generate this table. The approach, whose average value is larger, will be regarded as better. For example, users will have a higher chance of obtaining a recommendation using content-based approach while the accuracy of the recommended queries will be poor. The results-based approach improves average precision but suffers from poor coverage and recall. This result offers opportunities to enhance the performance of query clustering algorithms by using both query terms and the result URLs since the strength of individual approaches might balance the drawbacks of each other.

**Table 2.** Summary of the comparison between content-based and results-based approach

<table>
<thead>
<tr>
<th></th>
<th>Average cluster size</th>
<th>Coverage</th>
<th>Precision</th>
<th>Recall</th>
</tr>
</thead>
<tbody>
<tr>
<td>Better</td>
<td>Content-based approach</td>
<td>Content-based approach</td>
<td>Results-based approach</td>
<td>Content-based approach</td>
</tr>
<tr>
<td>Worse</td>
<td>Results-based approach</td>
<td>Results-based approach</td>
<td>Content-based approach</td>
<td>Results-based approach</td>
</tr>
</tbody>
</table>

Further, our experiments show that though the short length of queries might add doubt on the usefulness of the weight of terms, it does provide contributions to boosting the
coverage without damaging other metrics. Table 3 summarizes the comparison of different approaches in more detail, taking the impact of different thresholds into consideration. All the thresholds were categorized into two groups: low threshold, including 0.25 and 0.5, and high threshold, including 0.7 and 0.9. Note that for precision, sim_basic and sim_cosine generate similar results with regards to low threshold and high threshold respectively. This table was constructed based on the observation of previous figures, which further indicates the strength and weakness of different approaches based on different thresholds.

Table 3. Summary of comparison across all approaches

<table>
<thead>
<tr>
<th></th>
<th>Average cluster size</th>
<th>Coverage</th>
<th>Precision</th>
<th>Recall</th>
</tr>
</thead>
<tbody>
<tr>
<td>Low threshold</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(0.25, 0.5)</td>
<td>B</td>
<td>C</td>
<td>R</td>
<td>B</td>
</tr>
<tr>
<td></td>
<td>C</td>
<td>B</td>
<td></td>
<td>C</td>
</tr>
<tr>
<td></td>
<td>R</td>
<td>R</td>
<td>B &amp; C</td>
<td>R</td>
</tr>
<tr>
<td>High threshold</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(0.7, 0.9)</td>
<td>C</td>
<td>C</td>
<td>R</td>
<td>C</td>
</tr>
<tr>
<td></td>
<td>B</td>
<td>R</td>
<td></td>
<td>B</td>
</tr>
<tr>
<td></td>
<td>R</td>
<td>B</td>
<td>B &amp; C</td>
<td>R</td>
</tr>
</tbody>
</table>

B----sim-basic, C----sim-cosine, R----sim-result

6. Conclusions and Future Work

In this paper, we compare different query similarity measures. Our experiments show that by using content-based and results-based approaches alone, each method will generate drawbacks that will affect the quality of query clusters. From the results, it is obvious that the precision of 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. On the other hand, the results-based approach suffers from poor performance in terms of coverage while, this time, the content-based approach offers better results. Taking together, this indicates that the advantages of individual approaches offer opportunities to compensate the weakness points of each other. Therefore, they can complement each other in order to enhance the overall quality of final query clusters. Further, sim_cosine and sim_basic generate the similar results in almost all metrics except coverage, in which sim_cosine performs much better than sim_basic. This suggests that the weights of term can make contribution 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 experiment 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, such kind of system can help information seekers especially novices to express their information needs accurately.

In addition to the initial experiments performed in this research, experiments involving hybrid approaches, which exploit both query terms as well as result URLs, are also planned. Based on content-based and result URLs-based approaches, the hybrid approach might generate a balanced result than using them individually. Further, alternative approaches to identifying the similarity between queries will also be attempted. 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, word relationships like hypernyms can be used to replace query terms before computing the similarity between queries to increase the coverage as well as average cluster size. Finally, experiments using other clustering algorithms such as DBSCAN [2] might also be conducted to assess clustering quality. Since DBSCAN is a density-based clustering algorithm, it allows the system to find indirectly related queries besides the directly related queries for a given query. Hence, the average cluster size and coverage might be improved.

References


