Clustering User Queries of a Search Engine

Ji-Rong Wen
Microsoft Research, China
5F, Beijing Sigma Center
No.49, Zhichun Road Haidian District
Beijing, P.R.China
jrwen@microsoft.com

Jian-Yun Nie
Dept. Informatique et Recherche opérationnelle
University of Montreal
CP 6128, succursale Centre-ville
Montreal (Quebec), H3C 3J7 Canada
nie@IRO.Umontreal.CA

Hong-Jiang Zhang
Microsoft Research, China
5F, Beijing Sigma Center
No.49, Zhichun Road Haidian District
Beijing, P.R.China
hjzhang@microsoft.com

ABSTRACT

In order to increase retrieval precision, some new search engines provide manually verified answers to Frequently Asked Queries (FAQs). An underlying task is the identification of FAQs. This paper describes our attempt to cluster similar queries according to their contents as well as user logs. Our preliminary results show that the resulting clusters provide useful information for FAQ identification.

Categories and Subject Descriptors

Clustering, Citation and link analysis, Interactive IR

General Terms

Algorithms, Performance

Keywords

Query clustering, web data mining, user log, search engine

1. INTRODUCTION

The information explosion on the Internet has placed high demands on search engines. Yet people are far from being satisfied with the performance of the existing search engines, which often return thousands of documents in response to a user query. Many of the returned documents are irrelevant to the user's need. The precision of current search engines is well under people's expectations.

In order to find more precise answers to a query, a new generation of search engines - or question answering systems - have appeared on the Web (e.g. AskJeeves, http://www.askjeaves.com/). Unlike the traditional search engines that only use keywords to match documents, this new generation of systems tries to "understand" the user's question, and suggest some similar questions that other people have often asked and for which the system has the correct answers. In fact, the correct answers have been prepared or checked by human editors in most cases. It is then guaranteed that, if one of the suggested questions is truly similar to that of the user, the answers provided by the system will be relevant. The assumption behind such a system is that many people are interested in the same questions - the Frequently Asked Questions/Queries (FAQs). If the system can correctly answer these questions, then an important part of users' questions will be answered precisely. However, the queries submitted by users are very different, both in form and in intention. How human editors can determine which queries are FAQs is still an open issue. A closely related question is: how can a system judge if two questions are similar? The classic approaches to information retrieval (IR) would suggest a similarity calculation according to their keywords. However, this approach has some known drawbacks due to the limitations of keywords. At the retrieval level, traditional approaches are also limited by the fact that they require a document to share some keywords with the query to be retrieved. In reality, it is known that users often use keywords or terms that are different from the documents. There are then two different term spaces, one for the users, and another for the documents. How to create relationships for the related terms between the two spaces is an important issue. This problem can also be viewed as the creation of a live online thesaurus. The creation of such relationships would allow the system to match queries with relevant documents, even though they contain different terms. Again, with a large amount of user logs, this may be possible.

In this paper, we propose a new approach to query clustering using user logs. The principles are as follows. 1) If users clicked on the same documents for different queries, then the queries are similar. 2) If a set of documents is often selected for a set of queries, then the terms in these documents are related to the terms of the queries to some extent. These principles are used in combination with the traditional approaches based on query contents (i.e. keywords). Our preliminary results are very encouraging: many queries that we consider similar are actually clustered together using our approach. In addition, we notice that many similar questions would have been grouped into different clusters by traditional clustering approaches because they do not share any common keywords. This study demonstrates the usefulness of user logs for query clustering, and the feasibility of an automatic tool to detect FAQs for a search engine.

2. PROBLEM DESCRIPTION AND SUMMARY OF THE APPROACH

The current study has been carried out on the Encarta encyclopedia (http://encarta.msn.com/), which can be accessed on the Web. However, the approach described here may apply to most search engines.

Encarta contains a great number of documents organized in a hierarchy. A document is written on a specific topic, for instance, about a country. It is further divided into sections (and subsections) on different aspects of the topic. For example, there is usually a section about the population of a country, another section on its culture, etc. One should point out that the quality of the documents is consistently high, in contrast with other documents on the Web. The current search engine used on Encarta uses some advanced strategies to retrieve relevant documents. Although it suffers from some common problems of search engines, the result list does provide a good indication of the document contents. Therefore, when a user clicks on one of the documents in the result list, he/she has a good idea about what can be found in the document. This provides us with a solid basis for using user clicks as an indication of document's relevance.

Encarta is not a static encyclopedia, as are the printed ones. It evolves in time. In fact, a number of human editors are working on the improvement of the encyclopedia so that users can find more information from Encarta and in a more precise way. Their work mainly concerns the following two aspects: 1) If Encarta does not provide sufficient information for some often asked questions, the editors will add more documents to answer these questions. 2) If many users asked the same questions (FAQ) in a certain period of time (a hot topic), then the answers to these questions will be checked manually and directly linked to the questions. This second aspect is also found in many new-generation search engines, such as AskJeeves. Our current study concerns this aspect. The first goal is to develop a clustering system that helps the editors quickly identify the FAQs. The key problem is to determine an adequate similarity function so that truly similar queries can be grouped together using a clustering algorithm. Our second goal is to create a live online thesaurus that links the terms used by the users to those found in the documents, again through an analysis of user logs. Given a large amount of user logs, the relationships created could help the system match user queries and relevant documents despite differences in the terms they use.

2.1 Data

The available data is a large set of user logs from which we extracted query sessions. A session is defined as follows:

*session := query text [clicked document]**

Each session corresponds to one query and the documents the user clicked on. A query may be a well-formed natural language question, or one or more keywords or phrases. In Encarta, once a user query is input, a list of documents is presented to the user, together with the document titles. Because the document titles in Encarta are carefully chosen, they give the user a good idea of the contents of the documents. Therefore, if a user clicks on a document, it is likely that the document is relevant to the query, or at least related to it to some extent. Therefore, for our purposes, we do consider a clicked document to be relevant to the query. This assumption does not only apply to Encarta, but to most search engines. In fact, if among a set of documents provided by the system, the user chooses to click on some of them, it is because the user considers that these documents are more relevant than the others, based on the information provided in the document list (e.g. document title). Even if they are not all relevant, we can still affirm that they are usually more relevant than the other documents. In a long run, we can still extract interesting relationships from them.

The size of the query logs is very large. There are about 1 million queries per week that are submitted to Encarta. About half of query sessions have document clicks. Out of these sessions, about 90% of them have 1-2 document clicks. Even if some of the document clicks are erroneous, we can expect that most users do click on relevant documents.

2.2 Clustering Principles

Our approach is based on two criteria: one is on the queries themselves, and the other on user clicks.

The first criterion is similar to those used in traditional approaches to document clustering methods based on keywords. We formulate it as the following principle:

Principle 1 (using query contents): If two queries contain the same or similar terms, they denote the same or similar information needs.

Obviously, the longer the queries, the more reliable the principle 1 is. However, users often submit short queries to search engines. A typical query on the web usually contains one or two words. In many cases, there is not enough information to deduce users' information needs correctly. Therefore, the second criterion is used as a complement.

The second criterion is similar to the intuition underlying document clustering in IR. Classically, it is believed that closely associated documents tend to correspond to the same query [13]. In our case, we use the intuition in the reverse way as follows:

Principle 2 (using document clicks): If two queries lead to the selection of the same document (which we call a document click), then they are similar.

Document clicks are comparable to user relevance feedback in a traditional IR environment, except that document clicks denote implicit and not always valid relevance judgments.

The two criteria have their own advantages. In using the first criterion, we can group together queries of similar compositions. In using the second criterion, we benefit from user's judgments. This second criterion has also been used in [1] to cluster user queries. However, in that work, only user clicks were used. In our approach, we combine both user clicks and document and query contents to determine the similarity. Better results should result from this combination. We will discuss the use of these criteria in more detail in section 4.

2.3 Clustering Algorithm

Another question involved is the clustering algorithm proper. There are many clustering algorithms available to us. The main characteristics that guide our choice are the following ones:

1) The algorithm should not require manual setting of the resulting form of the clusters, e.g. the number of clusters. It is unreasonable to determine these parameters manually in advance.

2) Since we only want to find FAQs, the algorithm should filter out those queries with low frequencies.

3) Since query logs usually are very large, the algorithm should be capable of handling a large data set within reasonable time and space constraints.

4) Due to the fact that the log data changes daily, the algorithm should be incremental.

The density-based clustering method DBSCAN [2] and its incremental version Incremental DBSCAN [3] do satisfy meet the above requirements. DBSCAN does not require the number of clusters as an input parameter. According to the density-based definition, a cluster consists of the minimum number of points MinPts (to eliminate very small clusters as noise); and for every point in the cluster, there exists another point in the same cluster whose distance is less than the distance threshold Eps (points are densely located). The algorithm makes use of a spatial indexing structure (R*-tree) to locate points within the Eps distance from the core points of the clusters. All clusters consisting of less than the minimum number of points are considered as "noise" and are discarded. The average time complexity of the DBSCAN algorithm is O(n*logn). During our experiments, DBSCAN outperformed CLARANS [8] by a factor of between 250 and 1900, which increases with the size of the database. In our experiments, it only requires 3 minutes to deal with one-day user logs of 150,000 queries. Incremental DBSCAN is an incremental clustering algorithm, which can perform cluster updates on database incrementally. Due to the density-based nature of DBSCAN, the insertion or deletion of an object affects the current clustering only in the neighborhood of this object. Thus, efficient algorithms can be provided for incremental insertions and deletions within an existing cluster. In addition, based on the formal definition of clusters, it can be proven that the incremental algorithm yields the same results as DBSCAN. The performance evaluation of Incremental DBSCAN demonstrates its efficiency compared with the basic DBSCAN algorithm.

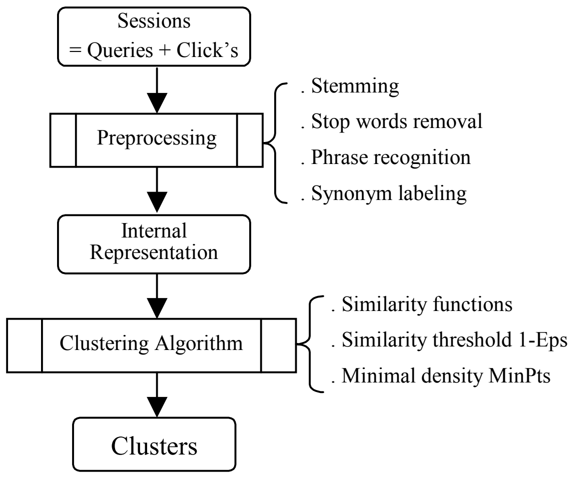


Figure 1. Flow chart of the clustering process.

We adopted DBSCAN and Incremental DBSCAN as the core algorithms to construct a comprehensive query clustering tool. This clustering tool is organized as shown in Figure 1.

One of the key problems is the choice of similarity function. In the remainder of the paper, we will focus on the calculation of similarity. Many studies have been carried out on this topic in the past. We will review some of them in the next section.

3. RELATED WORK ON SIMILARITY CALCULATIONS

The document clustering problem has been studied for a long time in IR [11]. Traditional approaches use keywords extracted from documents. If two documents share some keywords, then they are thought to be similar to some extent. The more they share common keywords, and the more these common keywords are important, the higher their similarity is. This same approach may also apply to query clustering, as a query may also be represented as a set of keywords in the same way as a document. However, it is well known that clustering using keywords has some drawbacks, due mainly to the fact that keywords and meanings do not strictly correspond. The same keyword does not always represent the same information need (e.g. the word "table" may refer to a concept in data structure or to a piece of furniture); and different keywords may refer to the same concept. Therefore, the calculated similarity between two semantically similar queries may be small, while two semantically unrelated queries may be considered similar. This is particularly the case when queries are short. In addition, in traditional IR methods, words such as "where" and "who" are treated as stopwords but are kept as keywords. For queries, however, these words encode important information about the user's need, particularly in the new-generation search engines such as AskJeeves. For example, with a "who"-question, the user intends to find information about a person.

Special attention is paid to such words in Query Answering (QA) [12] [5], where they are used as prominent indicators of question type. The whole question is represented as a template in accordance with the question type. The building of templates is crucial. In fact, one has to predefine a set of possible template forms for a given application. During question evaluation, the question template may be expanded using a thesaurus (e.g. Wordnet [7]) or morphological transformations. In our case, only a small portion of the queries are well-formed natural language questions. Most of queries are simply short phrases or keywords (e.g. "population of U.S."). The approach used in QA is therefore not completely applicable to our case. However, if words denoting a question type do appear in a complete question, these words should be taken into account.

Apart from document and query clustering based on keywords, there is still another group of approaches, which uses user's relevance feedback as an indication of similarity. The principle is as follows: if two documents are judged relevant to the same query, then there is reason to believe that these documents talk about some common topic, and therefore can be included in the same cluster. This approach to document clustering may solve some of the problems in using keywords, due to the implication of user judgments. However, in a traditional IR environment, the amount of relevance feedback information is too limited to allow for a reasonable coverage of documents. The situation in our case makes this approach feasible. In the Web environment, we have abundant queries and user clicks. As we mentioned previously, user clicks represent implicit relevance feedback. Although they are not as accurate as explicit relevance feedback and may even be erroneous, they do represent the fact the selected documents are thought of as being related to the query by the user. This forms the basis of our second clustering principle, which clearly distinguishes the current work from previous studies.

4. SIMILARITY FUNCTIONS

In this section, we will describe the similarity functions we use for our task.

4.1 Similarity Based on Query Contents

There are different ways to represent query contents: keywords, words in their order, and phrases. They provide different measures of similarity, each with its own useful information.

4.1.1 Similarity Based on Keywords or Phrases

This measure directly comes from IR studies. Keywords are the words, except for function words included in a stop-list. All the keywords are stemmed using the Porter algorithm [10]. The keyword-based similarity function is defined as follows:

$$similarity_{keyword}(p, q) = KN(p, q) / Max(kn(p), kn(q)) \quad (1)$$

where kn(.) is the number of keywords in a query, KN(p, q) is the number of common keywords in two queries.

If query terms are weighted, the following modified formula can be used instead:

$$similarity_{w\text{-}keyword}(p, q) =$$

$$\sum_{i=1}^{N} (w(k_i(p)) + w(k_i(q)))/Max(kn(p), kn(q)) * 2 \quad (2)$$

where $w(k_i(p))$ is the weight of the i-th common keyword in query p and $kn(.)$ becomes the sum of weights of the keywords in a query. In our case, we use *tf*idf* for keyword weighting.

The above measures can be easily extended to phrases. Phrases are a more precise representation of meaning than single words. Therefore, if we can identify phrases in queries, we can obtain a more accurate calculation of query similarity. For example, the two queries "history of China" and "history of the United States" are very close queries (both asking about the history of a country). Their similarity is 0.33 on the basis of keywords. If we can recognize "the United States" as a phrase and take it as a single term, the similarity between these two queries is increased to 0.5. The calculation of phrase-based similarity is similar to formulas (1) and (2). We only need to recognize phrases in a query. There are two main methods for doing this. One is by using a noun phrase recognizer based on some syntactic rules [6]. Another way is to use a phrase dictionary. In Encarta, there is such a dictionary. It contains a great number of phrases and proper nouns that appear in Encarta documents. This dictionary provides us with a simple way to recognize phrases in queries. However, it may not be complete. In the future, it will be supplemented by an automatic phrase recognizer based on a syntactic and statistical analysis.

4.1.2 Similarity Based on String Matching

This measure uses all the words in the queries for similarity estimation, even the stopwords. Comparison between queries becomes an inexact string-matching problem, as formulated by Gusfield [4]. Similarity may be determined by the edit distance, which is a measure based on the number of edit operations (insertion, deletion, etc.) necessary to unify two strings (queries). The similarity is inversely proportional to edit distance:

$$similarity_{edit}(p, q) = 1 - EditDistance(p, q) \quad (3)$$

The advantage of this measure is that it takes into account the word order, as well as words that denote query types such as "who" and "what" if they appear in a query. This method is more flexible than those used in QA systems, which rely on special recognition mechanisms for different types of questions. It is therefore more suitable to our situation.

In our preliminary experiments, we found that this measure is very useful for long and complete questions in natural language. Below are some queries that are grouped into one cluster:

Query 1: Where does silk come?

Query 2: Where does lead come from?

Query 3: Where does dew comes from?

This cluster contains questions of the form "Where does X come from?".

In the similarity calculations described above, we can further incorporate a dictionary of synonyms. Following Wordnet, we call a set of synonyms a synset. If two words/terms are in the same synset, their similarity is set at a predetermined value (0.8 in our current implementation). It is easy to incorporate this similarity between synonyms into the calculation of query similarity.

4.2 Similarity Based on User Feedback

Let *D(qi)* be the set of documents the system presents to the user as search results. The document set *D_C(.)* users clicked on for queries q_i and q_j may be seen as follows:

$$D\_C(q_i) = \{ d\_c_{i1}, d\_c_{i2}, \ldots, d\_c_{ip} \} \subseteq D(q_i)$$

$$D\_C(q_j) = \{ d\_c_{j1}, d\_c_{j2}, \ldots, d\_c_{jq} \} \subseteq D(q_j)$$

Similarity based on user clicks follows the following principle: If $D\_C(q_i) \cap D\_C(q_j) = \{ d\_c_{ij1}, d\_c_{ij2}, \ldots, d\_c_{ijt} \} \neq \varnothing$, then documents $d\_c_{ij1}, d\_c_{ij2}, \ldots, d\_c_{ijt}$ represent the common topics of queries q_i and q_j. Therefore, a similarity between queries q_i and q_j is determined by $D\_C(q_i) \cap D\_C(q_j)$.

There are two ways to consider Encarta documents: in isolation, or in terms of each document placed in a hierarchy (because of the hierarchical organization of documents in Encarta).

4.2.1 Similarity Through Single Documents

A first feedback-based similarity measure considers each document in isolation. This similarity is proportional to the shared number of clicked (or selected) documents, taken individually, as follows:

$$similarity_{document} = RD(p,q) / Max(rd(p), rd(q)) \quad (4)$$

where *rd*(.) is the number of clicked documents for a query, *RD*(*p,q*) is the number of document clicks in common.

In spite of its simplicity, this measure demonstrates a surprising ability to cluster semantically related queries that contain different words. Below are some queries from one such cluster:

Query 1: atomic bomb

Query 2: Nagasaki

Query 3: Nuclear bombs

Query 4: Manhattan Project

Query 5: Hiroshima

Query 6: nuclear fission

Query 7: Japan surrender

……

They all correspond to the document "ID: 761588871, Title: Atomic Bomb" in Encarta.

In addition, this measure is also very useful in distinguishing between queries that happen to be worded similarly but stem from different information needs. For example, if one user asked for "law" and clicked on the articles about legal problems, and another user asked "law" and clicked the articles about the order of nature, the two cases can be easily distinguished through user clicks. This kind of distinction can be exploited for sense disambiguation in a user interface. This is an aspect we will investigate in the future.

4.2.2 Similarity Through Document Hierarchy

Documents in Encarta are not isolated; they are organized into a hierarchy that corresponds to a concept space. The hierarchy contains 4 levels. The first level is the root. The second level contains 9 categories, such as "physical science & technology", "life science", "geography", etc. These categories are divided into 93 subcategories. The last level (the leaf nodes) is made up of tens of thousands of documents. This concept hierarchy allows us extend the previous calculation by considering a conceptual distance between documents. This distance is determined as follows: the lower the common parent node two documents have, the shorter the conceptual distance between the two documents. Let *F(di, dj)* denote the lowest common parent node for documents *di* and *dj*, *L(x)* the level of node *x*, *L_Total* the total levels in the hierarchy (i.e. 4 for Encarta). The conceptual similarity between two documents is defined as follows:

$$s(d_i, d_j) = (L(F(d_i, d_j)) - 1) / (L\_Total - 1) \qquad (5)$$

In particular, $s(d_i, d_i) = 1$; and $s(d_i, d_j) = 0$ if $F(d_i, d_j) =$ root.

Now let us incorporate this document similarity measure into the calculation of query similarity. Let d_i $(1 \le i \le m)$ and d_j $(1 \le j \le m)$ be the clicked documents for queries *p* and *q* respectively, and *rd(p)* and *rd(q)* the number of document clicks for each query. The hierarchy-based similarity is defined as follows:

$$similarity_{concept}(p, q) = \frac{\sum_{i=1}^{m} (\max_{j=1}^{n} s(d_i, d_j)) \Big/ rd(p) + \sum_{j=1}^{n} (\max_{i=1}^{m} s(d_i, d_j)) \Big/ rd(q)}{2} \qquad (6)$$

Using formula (6), the following two queries are recognized as being similar:

Query 1: <query text> image processing

<clicked documents> ID: 761558022 Title: Computer Graphics

Query 2: <query text> image rendering

<clicked documents> ID: 761568805 Title: Computer Animation

Both documents have a common parent node "Computer Science & Electronics". According to formula (5), the similarity between the two documents is 0.66. If these were the only two documents selected for the two queries, then the similarity between the queries is also 0.66 according to formula (6). In contrast, their similarity based on formula (4) using common clicks is 0. Hence, we see that this new similarity function can recognize a wider range of similar queries.

4.3 Combination of Multiple Measures

Similarities based on query contents and user clicks represent two different points of view. In general, content-based measures tend to cluster queries with the same or similar terms. Feedback-based measures tend to cluster queries related to the same or similar topics.

Since user information needs may be partially captured by both query texts and relevant documents, we would like to define a combined measure that takes advantage of both strategies. A simple way to do this is to combine both measures linearly, as follows:

$$similarity_{comp} = \alpha * similarity_{content} + \beta * similarity_{feedback} \qquad (7)$$

This raises the question of how to set parameters α and β. We believe that these parameters should be set according to editor's objectives. It is difficult to determine them in advance; they can be adjusted over time and in light of the system's use. In what follows, we give a simple example to show the possible effects of different measures, as well as their combination.

Let us consider the 4 queries shown in Table 1. Assume that the similarity threshold is set at 0.6. The ideal result would be to group Queries 1 and 2 in a cluster, and Queries 3 and 4 in another.

Table 1. Some query examples.

Query 1: <query text> law of thermodynamics	
<clicked documents>	ID: 761571911 Title: Thermodynamics
	ID: 761571262 Title: Conservation Laws
Query 2: <query text> conservation laws	
<clicked documents>	ID: 761571262 Title: Conservation Laws
	ID: 761571911 Title: Thermodynamics
Query 3: <query text> Newton law	
< clicked documents>	ID: 761573959 Title: Newton, Sir Isaac
	ID: 761573872 Title: Ballistics
Query 4: <query text> Newton law	
< clicked documents>	ID: 761556906 Title: Mechanics
	ID: 761556362 Title: Gravitation

Table 2. Similarities between documents

	①	②	③	④	⑤	⑥
①Thermodynamics	1.0	0.66	0.33	0.33	0.66	0.66
②Conservation laws	0.66	1.0	0.33	0.33	0.66	0.66
③Newton, Sir Isaac	0.33	0.33	1.0	0.33	0.33	0.33
④Ballistics	0.33	0.33	0.33	1.0	0.33	0.33
⑤Mechanics	0.66	0.66	0.33	0.33	1.0	0.66
⑥Gravitation	0.66	0.66	0.33	0.33	0.66	1.0

1) If the keyword-based measure is applied (formula (1)), the queries are divided into 3 clusters:

   Cluster 1: Query 1

   Cluster 2: Query 2

   Cluster 3: Query 3 and Query 4

   Queries 1 and 2 are not clustered together.

2) If we use the measure based on individual documents (formula (4)), we obtain:

   Cluster 1: Query 1 and Query 2

   Cluster 2: Query 3

   Cluster 3: Query 4

   Now Queries 3 and 4 are not judged to be similar.

3) Now we use the measure on document hierarchy. The document similarities according to formula (5) are shown in table 2.

   Applying formula (6), we can group the queries as follows:

   Cluster 1: Query 1, Query 2, and Query 4

   Cluster 2: Query 3

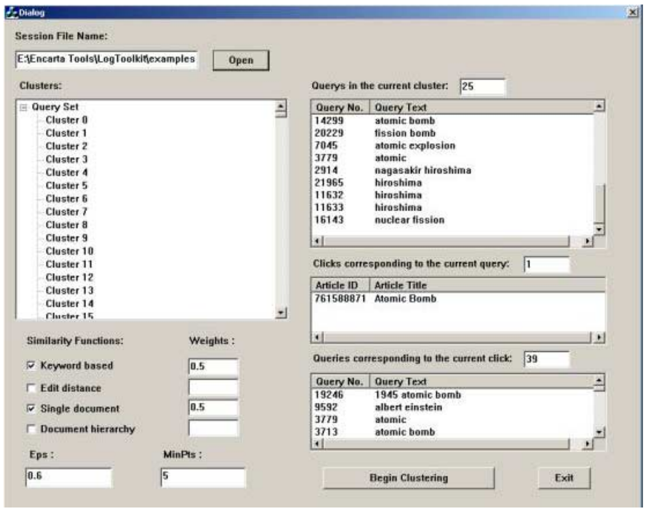
   We see that it is not possible to separate Query 4 from Queries 1 and 2.

4) Now let us use formula (7) with $\text{similarity}_{content}$ = $\text{similarity}_{keyword}$, and $\text{similarity}_{feedback}$ = $\text{similarity}_{concept}$. Both α and β are set to 0.5. The queries are now clustered in the desired way:

   Cluster 1: Query 1 and Query 2

   Cluster 2: Query 3 and Query 4

The purpose of this example is only to show that each similarity calculation focuses on a specific aspect. By combining them in a reasonable way, better results can be obtained. Therefore, by trying different combinations, the editors will have a better chance of locating desired FAQs. Our current implementation includes all the similarity measures described above. An interface allows the editors to choose different functions and to set different combination parameters (see figure 2).



Session File Name: E:\Encarta Tools\LogToolkit\examples

Clusters:

Query Set
- Cluster 0
- Cluster 1
- Cluster 2
- Cluster 3
- Cluster 4
- Cluster 5
- Cluster 6
- Cluster 7
- Cluster 8
- Cluster 9
- Cluster 10
- Cluster 11
- Cluster 12
- Cluster 13
- Cluster 14
- Cluster 15

Similarity Functions: / Weights :

☑ Keyword based	0.5
☐ Edit distance	
☑ Single document	0.5
☐ Document hierarchy	

Eps : 0.6 MinPts : 5

Querys in the current cluster: 25

Query No.	Query Text
14299	atomic bomb
20229	fission bomb
7045	atomic explosion
3779	atomic
2914	nagasakir hiroshima
21965	hiroshima
11632	hiroshima
11633	hiroshima
16143	nuclear fission

Clicks corresponding to the current query: 1

Article ID	Article Title
761588871	Atomic Bomb

Queries corresponding to the current click: 39

Query No.	Query Text
19246	1945 atomic bomb
9592	albert einstein
3779	atomic
3713	atomic bomb

Figure 2. Interface of the query clustering tool

Obviously, we need more empirical evidence to be able to affirm the effectiveness of our query clustering algorithms, especially the various similarity functions. So far, we have run the system on daily user logs of the Encarta encyclopedia website. Encarta content editors and indexers are using this query clustering tool to identify FAQs and the topics and information that users want the most. This allows a small editorial team to focus on improving the content and organization of the information that users are most frequently after. Their first conclusion is that the system does cluster queries appropriately, although there is certainly room for further improvement. Currently, we are carrying out additional experiments to formally evaluate various aspects of our query clustering methods, such as the characteristics of different similarity functions, the quality of clustering results, weight setting with respect to combined measures, etc. We will report on our experimental results in future publications.

5. CONCLUSION

The new generation of search engines for precise question answering requires the identification of FAQs, so that human editors may prepare the correct answers to them. The identification of FAQs is not an easy task; it requires a proper estimation of query similarity. Given the different forms of queries and user intentions, the similarity of queries cannot be accurately estimated through an analysis of their contents alone (i.e. via keywords). In this paper, we have suggested exploiting user log information (or user document clicks) as a supplement. A new clustering principle is proposed: if two queries give rise to the same document clicks, they are similar. Our initial analysis of the clustering results suggests that this clustering strategy can effectively group similar queries together. It does provide effective assistance for human editors in discovering new FAQs.

Our project is still ongoing, and there is certainly room for further improvement:

1) The identification of phrases may be extended by using an automatic recognizer based on syntax.

2) The calculation of query similarity could incorporate more relationships between terms, besides the synonyms stored in the Encarta dictionary, perhaps by a co-occurrence analysis on documents in Encarta. Another possible solution is to extract relationships between query terms and the terms in the clicked documents via a statistical analysis.

3) The system could be extended into a word disambiguation tool by considering each clicked document as a possible meaning of a query word.

In summary, the availability of large amounts of user logs is new with respect to the traditional IR environment, and they suggest new possibilities for search engines. In particular, these logs are highly informative for the analysis of trends in user searches, which can help search engine builders and editors to improve their systems. The present study is only a first step in this direction.

REFERENCES

[1] Beeferman, D. and Berger. A. Agglomerative clustering of a search engine query log. KDD'2000, 2000, pp. 407-416.

[2] Ester, M., Kriegel, H., Sander, J. and Xu, X. A density-based algorithm for discovering clusters in large spatial databases

with noise. Proc. 2nd Int. Conf. on Knowledge Discovery and Data Mining. Portland, OR. 1996, pp. 226-231.

[3] Ester, M., Kriegel, H., Sander, J., Wimmer, M. and Xu, X. Incremental clustering for mining in a data warehousing environment. Proc. of the 24th VLDB Conf. New York, USA, 1998.

[4] Gusfield, D. Algorithms on strings, trees, and sequences: computer science and computational biology. Part III. "Inexact matching, sequence alignment, and dynamic programming". Press of Cambridge University. 1997.

[5] Kulyukin, V.A., Hammond, K.J. and Burke, R.D. Answering questions for an organization online, Proc. of AAAI'98, 1998.

[6] Lewis, D.D. and Croft, W.B. Term clustering of syntactic phrases, ACM-SIGIR, pp. 385-404, 1990.

[7] Miller, G. (Ed.) Wordnet: an on-line lexical database. International Journal of Lexicography. 1990.

[8] Ng, R. and Han, J. Efficient and effective clustering method for spatial data mining. In Proc. 1994 Int. Conf. Very Large Data Bases, pp. 144--155, Santiago, Chile, September 1994.

[9] Robertson, S.E., Maron, M.E. and Cooper, W.S. Probability of relevance: a unification of two competing models for document retrieval. Information Technology: Research and Development, 1(1-21). 1982.

[10] Porter, M. An algorithm for suffix stripping. Program, Vol. 14(3), pp. 130137, 1980

[11] Salton, G. and McGill, M.J. Introduction to modern information retrieval, McGraw-Hill Book Company, 1983.

[12] Srihari, R. and Li, W. Question answering supported by information extraction. Proc. of TREC8, pp. 75-85, 1999.

[13] van Rijsbergen, C.J. Information Retrieval. Butterworths, 1979.