Query Recommendation in Digital Libraries using OLAP[*]

Carlos Garcia-Alvarado
University of Houston
Dept. of Computer Science
Houston, TX, USA

Carlos Ordonez
University of Houston
Dept. of Computer Science
Houston, TX, USA

Zhibo Chen
University of Houston
Dept. of Computer Science
Houston, TX, USA

ABSTRACT

Query suggestion is well-known to enhance the user's search for relevant documents. In this work, we propose a novel technique that emulates a human skill when searching or exploring digital collections. In general, a user begins searching by providing a naïve query and then analyzes the retrieved documents in order to refine the query search. We decided to emulate this behavior by generating alternative queries using OLAP. Such queries are the result of performing multiple data summarizations on digital libraries, and then generating cuboids depending on the correlation between the keywords of the collection and the subset of keywords belonging to the previous search. Moreover, we introduce techniques to efficiently obtain query suggestions inside the DBMS by exploiting UDFs and SQL queries.

Categories and Subject Descriptors

H.2.8 [**Information Systems**]: Database applications

General Terms

Algorithms, Experimentation

Keywords

OLAP, IR, Query Processing

1. INTRODUCTION

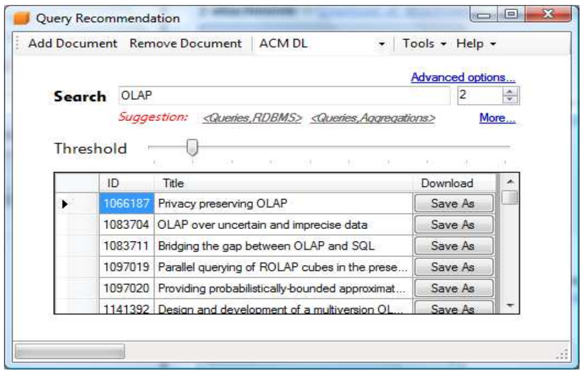
Exploring and searching large digital collections such as the ACM Digital Library, IEEE Xplore, or even metadata collections such as DBLP, have become cumbersome when the user possesses only incomplete facts or is missing information for properly choosing the keywords to search [1]. However, as the user performs a certain number of searches, he begins to realize the keywords needed for obtaining the desired information by analyzing the retrieved documents. As a result, the user will eventually arrive at his target query, which returns the desired information. For example, "data summarization" may be our initial naïve search, but eventually, through a set of search iterations, we realize that the proper subset of documents is not really about "data summarization," but rather "data cubes." Following this behavior, we believe that an initial naïve query is not independent of the target query. Consequently, we emulated this searching behavior by providing the user with a set of query suggestions that are the result of analyzing the frequency of the keywords of a subset of the documents per query.

[*]This research work was partially supported by NSF grants CCF 0937562 and IIS 0914861.

2. TECHNICAL DETAIL

Our application relies on a ranking model for retrieving the top-k documents after an initial search by the user. Thereafter, the recommendation is obtained by analyzing this subset of documents. An additional step during the preprocessing phase includes the computation of the correlation matrix between keywords. The keywords with frequencies higher than a certain value, as defined by the user, are selected, and by using the keyword-document pairs, the final correlation matrix is obtained. Hence, the total frequency of each term in the collection must be computed, resulting in the most costly step in the application. However, the correlation matrix is only computed once per collection, and can also be computed incrementally when new keywords are added. Additionally, to compute a measurement between more than two keywords, we multiplied all the related coefficients to obtain a discrimination measure.

Also, this recommendation tool also relies on the summarization properties of OLAP[4]. We perform OLAP on top of a DBMS without the need to alter any of the underlying source code or use a separate OLAP Server. For our research, we use OLAP to generate a keyword data cube in which each node of the lattice represents one specific combination of keywords (d). The data cube stores both the number of occurrences of the combination in the document space, as well as the total number of documents in which the set of keywords appears (e). For example, suppose the node "Data Cubes" stores the values 150 (frequency) and 10 (number of documents). Then we know that in our filtered document space, this combination of keywords appears a total of 150 times in 10 different documents. Thus, the p-th level of the lattice would contain all possible combinations of p keywords. For keywords, the dimensionality is often too high to generate the entire lattice at once, and we will certainly run out of resources and time. Moreover, presenting all the possible combinations of keywords would overwhelm the user with too much information at once. As a result, we perform a user-driven traversal of the dimensional lattice. In this process, we first provide the user with a set of keywords in a p level, which is selected by the user (e.g. level


Query Recommendation

Add Document | Remove Document | ACM DL | Tools | Help

Advanced options...

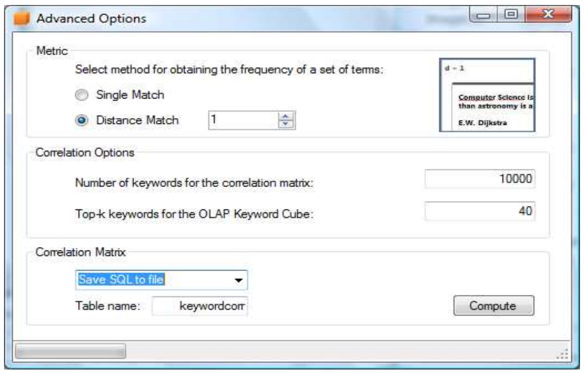
Search: OLAP — 2

Suggestion: <Queries,RDBMS> <Queries,Aggregations> More...

Threshold

ID	Title	Download
1066187	Privacy preserving OLAP	Save As
1083704	OLAP over uncertain and imprecise data	Save As
1083711	Bridging the gap between OLAP and SQL	Save As
1097019	Parallel querying of ROLAP cubes in the prese...	Save As
1097020	Providing probabilistically-bounded approximat...	Save As
1141392	Design and development of a multiversion OL...	Save As

Figure 1: Query Search.


Advanced Options

Metric

Select method for obtaining the frequency of a set of terms:

Single Match [✓ Distance Match] 1

Correlation Options

Number of keywords for the correlation matrix: 10000

Top-k keywords for the OLAP Keyword Cube: 40

Correlation Matrix

Save SQL to file

Table name: keywordcorr

Compute

Figure 2: OLAP Keyword Cube Parameters.

2 of the dimensional lattice will return all the pairs of keywords), found within the document space. From here, the user can activate further calculations by drilling down the lattice. For example, suppose the user wants to drill into the keywords “Information Retrieval”. Then our application would calculate only those nodes involving these two keywords at the next level to form sets of three keywords, such as “Information Retrieval Query”. Hence, we avoid computing large portions of the lattice at once while still providing the user with the flexibility to drill as far into the lattice as needed. Currently, we have explored two ways in which we can execute OLAP queries: a pure SQL approach and a User-Defined Function (UDF) approach. Further details regarding these two methods can be found in [2].

The computation of the frequency of a set of keywords in a document can be obtained by using two different approaches: single match or distance match. While both are considered an independent level of the cube, each metric returns results of differing qualities. Single match is the result of considering a keyword set occurrence as the result of an appearance of all the keywords in the set independently. Thus, if keyword1 appears three times in the document and keyword2 appears one time in the same document, the combination {keyword1,keyword2} is considered to have a frequency of one. Distance match, on the other hand, takes into consideration the distance between both keywords in a document (the distance is given by the user). However, using the distance method can be troublesome to adjust. If the distance is too small, the cube may not have enough occurrences to aggregate; if the distance is too large, the final quality of the results may be equivalent to those in the single match without the overhead of the distance computation.

3. SYSTEM DEMONSTRATION

We developed our system as a standalone application using the Vector Space Model for retrieving the top-k documents and connecting to the DBMS via ODBC. The user interface is a thin client for sending and retrieving SQL statements, and the core of the computations is performed in the DBMS. Traditional data structures for supporting IDF-TF are built inside the DBMS to allow the ranking of the documents (see [3]). In this demonstration scenario, our goal is to show how we can explore four different collections of documents from the UCI Repository (ENRON, KOS, NIPS, NYTIMES), and an additional collection of documents related to Water Pollution from the Texas Commission on Environmental Quality. The first set of collections will be used exclusively for the single match metric, and the Water Pollution collection will be used to test both metrics. The fact that they can be navigated by a series of suggestions given by the applications following the content of the users' queries will be easily appreciated. The demonstration will be divided into two sections: the preprocessing phase, and the retrieval and exploration phase. The preprocessing phase (see Figure 2) will show the creation of the correlation matrix of a small collection, due to the time constraint for showing a larger collection. The construction of the required data structures will be shown by displaying the SQL employed for the keyword preprocessing stage and the lineal algebra operations required to obtain the final results. The second phase of the demonstration will be focused on searching and mostly on the query suggestion. The users will have the opportunity to select the collection on which they would like to perform the query. The query recommendation will require setting parameters such as the threshold for the iceberg queries, and the size of the combination of keywords used in the keyword suggestion (see Figure 1). Then a set of naïve queries will be given, and the audience will have the chance to explore the collection by choosing from given suggestions. Finally, the users will appreciate that a such query suggestion is the result of an on-the-fly OLAP Cube computation by using in-database extensions and SQL queries to explore and query digital collections. In addition, the user will experience that using the OLAP query suggestions allows a clever navigation of all of the keywords associated with an initial query, in order to find the target query that displays the desired results. This is a capability that it is not possible when performing a simple search in a collection.

4. REFERENCES

[1] Y. Chen, W. Wang, and X. Lin Z. Liu. Keyword Search on Structured and Semi-structured Data. Tutorial Description. In *SIGMOD*, 2009.

[2] Z. Chen, C. Ordonez, and C. Garcia-Alvarado. Fast and Dynamic OLAP Exploration Using UDFs. In *SIGMOD*, pages 1087–1090, 2009.

[3] C. Garcia-Alvarado and C. Ordonez. Information retrieval from digital libraries in SQL. In *WIDM Workshop*, pages 55–62, 2008.

[4] C. Ordonez and Z. Chen. Evaluating Statistical Tests on OLAP Cubes to Compare Degree of Disease. *IEEE TITB*, 13(5):756–765, 2009.