On the Design and Evaluation of a Multi-dimensional Approach to Information Retrieval

M. Catherine McCabe
U.S. Government
Washington, DC 20505
catherm @ ir.iit.edu

Jinho Lee, Abdur Chowdhury, David Grossman, Ophir Frieder
Illinois Institute of Technology
Chicago, IL 60616
{jinho, abdur, dagr, ophir} @ ir.iit.edu

Abstract

We present a method of searching text collections that takes advantage of hierarchrical information within documents and integrates searches of structured and unstructured data. We show that Multidimensional databases (MDB), designed for accessing data along hierarchical dimensions, are effective for information retrieval. We demonstrate a method of using On-Line Analytic Processing (OLAP) techniques on a text collection. This combines traditional information retrieval and the slicing, dicing, drill-down, and roll-up of OLAP. We demonstrate use of a prototype for searching documents from the TREC collection.

1 Introduction

Text documents often contain references to hierarchical information. An organization mentioned in text has a location, a CEO, an industry, etc. Similar information exists for persons. In addition, the document may have metadata such as publisher, date, author, etc. We propose a technique for making this structured information available to searches using an On-Line Analytic Processing (OLAP) tool. We present how text is modeled and accessed in a Multidimensional database (MDB).

With our approach, users may narrow or broaden the search along hierarchical lines. For example the query – *Find documents published at this LOCATION* – may be narrowed or broadened by city, state, etc. For a particular location, the results can be further narrowed i.e. *how many mention this PERSON?* PERSON could be another dimension with associated information such as employer. The user can then jump to *Find documents about people with the same employer as PERSON.* Such dynamic movement through data is a typical OLAP feature and well suited to text analysis.

We implemented our approach using MS OLAP Services using a subset of 1,827 TREC documents. Future work includes scaling to larger collections.

SIGIR 2000 7/00 Athens, Greece
© 2000 ACM ISBN 1-58113-226-3/00/0007...$5.00

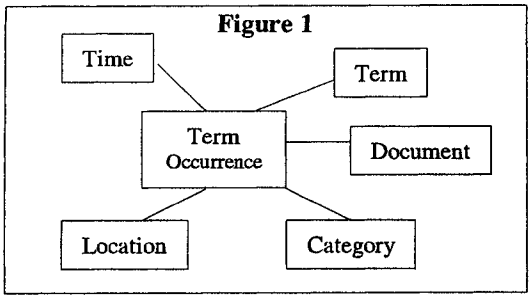
2 Prior Work

OLAP and MDB effectively analyze large collections of structured data [1]. Likewise, Information Retrieval (IR) succeeds in searching unstructured text and returning ranked lists of documents [2, 3, 4]. Finally, work exists integrating searches of structured and unstructured data [5, 6, 7]. However those efforts do not take advantage of the hierarchical nature of structured data nor of hierarchies in the text. Multidimensional IR makes use of such hierarchies and allows the user a new kind of interface for analyzing sets of documents [8].

3 Data Model

A multi-dimensional data model for text permits the definition of any dimension of interest. The dimensions might be hierarchies or simply related data. For example, known geographical hierarchies (city, state, country) could be reference data loaded into a dimension. A document might contain a term from any level of the hierarchy (i.e. Illinois). Now the OLAP tool permits the document to be accessed by city or country even though they were not explicitly mentioned in the document. One might use the presence of terms to match the reference data. However, this is prone to much ambiguity error. A more certain match is obtained when the documents have tags to confirm the dimension. For example, the FR documents have a <LOCATION> tag indicating the location of publication. In addition, the use of an extraction tool provides further identification of dimensions within documents. Such commercial products use Natural Language Processing (NLP) and rules to identify *entities* within documents. For example, the NetOwl Extractor tags PERSON, PLACE, and ENTITY (organizations) and identifies some hierarchical information about each. By using such tagging, we can ensure that Saint Michael the person, the organization (a church) and the location (city) are not mismatched. Given additional reference data such as an employee database, we can now match documents to hierarchies such as employer, department, etc.

Typically, star schemas are used to model multidimensional data [9]. A star schema for text documents is shown in Figure 1.



The central table is the TERM_OCCURENCE. There is a row here for each term occurrence in each document. This gets quite large for large document collections. It is the central fact table and thus contains a key to each dimension -- Time Id, Location Id, Category Id, Term Id, Doc Id, and Weight. The weight is a value for this term within this document, such as term frequency. We used the more complex pivoted cosine weight [10], although any term-weighting scheme could be used.

We defined several dimensions for our collection – TIME, LOCATION, TERM, DOCUMENT, and CATEGORY. In our schema, TIME, and LOCATION are hierarchies while the DOCUMENT dimension contains structured, bibliographic information. The TERM dimension is defined as <term, category, term-weight>. Term-weight is a value for this term across the collection such as inverse document frequency (*idf*). It may be used in conjunction with the weight in TERM_OCCURANCE to calculate relevance (i.e. tf-idf). The CATEGORY dimension might contain subject category for the term or the part-of-speech (duck as a noun or as a verb) or a hierarchy such as Wordnet.

4 Document Analysis & Search Prototype

We used MS OLAP Services to build the cube modeled in Figure 1. Just as standard SQL is used to access relational DBMS, multidimensional expressions (MDX) is a growing de facto standard for MDB [11]. We populated the cube by loading flat files generated from preprocessing the text (parsing, removing stop words, etc.) Now, we run MDX queries.

4.1 Document Search

First, we examine the use of OLAP for traditional IR. The MDX query shown in Example 1 implements IR based on the vector space model. To represent a query, we create QUERYTERM as another dimension in the cube. It consists of all the terms in the query.

The MDX in Example 1 calculates relevance ranking by summing the term_weights of the query terms within each document. Note that the summing is not explicit but rather the natural result of the roll-up feature of OLAP. Since we selected *docid*, and *weight* and the WHERE clause calls for all documents containing query terms, the results are rolled-up across all query terms in the document This is analogous to a query on a TIME hierarchy requesting data for all of 1999 instead of just the first quarter. Finally, the TopCount function results in displaying only the top *n* relevant documents (10, in example 1).

```
SELECT
  NON EMPTY { [measures].weight } on columns,
  NON EMPTY TopCount (Order(
  [doc_Id].members, [measures].weight, DESC), 10)
  on rows
FROM docInfo WHERE  [queryterm]
```

Example 1: Vector Space Relevance Ranking

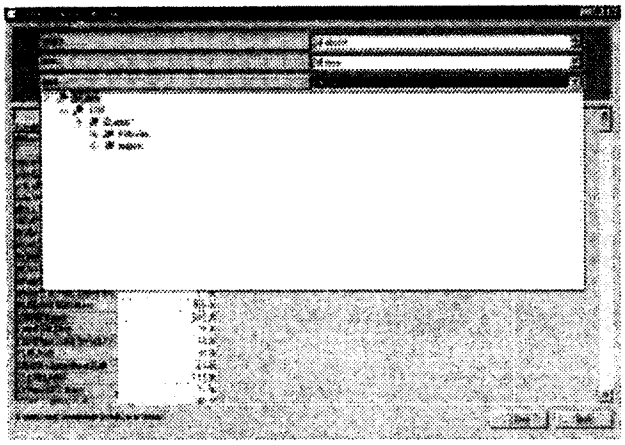
4.2 Analysis – Slicing and Dicing

In addition to traditional document searches, the OLAP platform provides an environment for analyzing a set of documents. Simple pull-down menus list available dimensions and users may view the documents along any dimension. We processed a subset of Wall Street Journal articles from the TREC collection into the system. Example 2 shows the MDX query for integrating the hierarchical DATE and LOCATION with a term-based query. The query is: "Find all the documents on *forests* published in New York in the first quarter of 1998."

```
SELECT
  NON EMPTY [DocId].members on rows,
  {[Measures].[Tf] } on columns
FROM DocInfo
  WHERE ([Term].[forest], [1998][quarter 1],
    [Location].[New York])
```

Example 2: Hierarchical search of Location & Time

During an interactive session, the user pivots from one question to another, walking up and down hierarchies and dynamically changing the set being reviewed. With a pull-down menu, the user asks the question, *"When were documents published from this location?"* The TIME dimension is invoked. Display 1 shows the user changing the TIME dimension under review. When a pivot is done from this screen, the user jumps from looking at all documents published at this location, for this month, to all documents published this month, for any location.



Display 1: Changing Time Dimension

4.3 Testing on TREC documents

We evaluated the IR searching function of our prototype by using the relevance assessments provided by NIST for the TREC collection. We processed 1,827 Federal Register documents from the corpus. This subset was selected due to the density of relevant documents. We ran fourteen TREC7 queries that contained relevant documents in our collection. Finally, we evaluated the average precision of our retrieval.

The fourteen queries achieved an average precision recall of 0.4261 on Title queries. We note that as the collection is small and the number of relevant documents per query is small, this measure is simply a rough indication retrieval success, and not a predictor of retrieval against larger collections. However, the value matches that achieved with inverted index approaches and verifies the success of this environment for IR.

The benefit of using this environment for retrieval is that the end-user may browse through the result sets, scanning through based on terms, structured data or hierarchical information. For example, Query 370 requests documents on 'food/drug laws'. There are 65 relevant documents for this in our collection. Since these are Federal Register documents, many of them contain HTML tags for <DEPARTMENT> and <AGENCY> where AGENCY is a sub-organization within DEPARTMENT. In addition, many of the documents contain tags for <SIGNER> and <SIGNJOB>, which are the name of the person who signed the document, and the position of that person. Given the right cube design, the end-user may jump from viewing all the documents, to those from FDA, view the SIGNERs of those documents and then jump to all the documents by given SIGNER (even before he worked for FDA.) While an IR system could be built to allow the users to query along these hierarchies, the OLAP tool gives the advantages of precalculated data cubes designed for fast access.

5 Discussion and Future Work

We have demonstrated the use of OLAP techniques, specifically MDB's, as an approach to information retrieval. By modeling the text and data hierarchies in a star schema, searchers gain the advantage of typical OLAP -- slicing and dicing, drilling down and up. This adds significantly to the analysis and understanding of the documents that return from a text search. Users interactively explore the documents returned from their queries and potentially gain insight much more quickly than by reading through a ranked list of documents.

Future work includes scaling the existing prototype to larger document collections, optimizing the schema and developing special access structures for performance and experimentation with sparse cube operations needed for text application.

References

[1] M. Gyssens and V.S. Lakshmanan. A Foundation for Multi-Dimensional DB. *Proceedings of the 23rd Very Large DataBase Conference.* 1997.

[2] G. Salton, C.S. Yang and A. Wong. A vector-space model for IR, *Comm. of the ACM*, 18, 1975.

[3] D. Grossman and O. Frieder. Information Retrieval: Algorithms and Heuristics. Kluwer, 1998.

[4] Proceedings of the Text Retrieval Evaluation Conference TREC8, by NIST and ARPA 1999.

[5] D. Grossman, D. Holmes, O. Frieder and D. Roberts, Integrating Structured Data and Text: A Relational Approach, *JASIS*, February 1997.

[6] C. Lynch and M. Stonebraker. Extended user-defined indexing with application to textual database. In *Proceeding of the 14th VLDB Conf.*, p. 306-317, 1988.

[7] I. Macleod. Sequel as a language for document retrieval. *JASIS*, Pages 243-249, September 1979.

[8] J. Lee, D. Grossman, O. Frieder, M. C. McCabe. Integrating Structured Text and Data: A Multidimensional Approach. *Proc. Of ITCC2000.* March 2000.

[9] R. Kimball, Data Warehouse Toolkit, John Wiley & Sons, '96.

[10] A. Singhal, C. Buckley, and M. Mitra, "Pivoted Document Length Normalization," *In Proc. of SIGIR* 96 Ed. H-P. Frei, D. Harman, P. Schauble, and R. Wilkinson, 1996.

[11] Advanced MDX query available at http://msdn.microsoft.com/library/sdkdoc/dasdk/olap8rg8.htm.