Generic Search Optimization for Heterogeneous Data Sources

Majid Zaman
Scientist, Directorate of Information Technology & Support Systems, University of Kashmir, Srinagar, J&K, India

S. M. K. Quadri
Head & Director, PG Department of Computer Science, University of Kashmir, Srinagar, Srinagar, J&K, India

Muheet Ahmed Butt
Scientist, Directorate of Information Technology & Support Systems, University of Kashmir, Srinagar, J&K, India

ABSTRACT

Data Retrieval is still a pervasive challenge faced in applications that need to query across multiple autonomous and heterogeneous data sources. There is decent amount of standardization as far as World-Wide Web is concerned, while google is universal access tool to search and determine source of the information user requires there is still no such tool that can be implemented at enterprise level where there are multitude of data sources and organization users are still facing difficulty in accessing data available on the intranet of the organization and not on the WWW, in order to access such data users within the organizations need to know a lot including location, access techniques etc while still data consistency & redundancy is beyond the scope of common organization user/s.

This paper introduces GENERIC SEARCH PRINCPLE: Solution making use of Knowledge base where in users of the organization irrespective of their technical ability, data source knowledge and location can search heterogeneous data sources including legacy data sources of organization and retrieve information, also taking into consideration user attributes like his/her location, work profile, designation etc so as to make search more relevant and results more precise.

KEYWORDS

Expert System, data sources, metadata, Knowledge Base

1. INTRODUCTION

The Success of World Wide Web resulted in massive increase in amount of data made public over the web[1], however complex problems arise when the result is to be generated/combined from n heterogeneous[3][10][15][16] data sources example being piece of data to be searched can be available in m data sources and in k different formats, thus obtaining information from n sources can yield inconsistent/contradictory values/results.

With almost every enterprise ranging from small medium to large deploying database applications for efficient storage and retrieval of process specific data, the databases have grown out in volume and have much more data. The optimum utilization of this data will happen only when every end user can get the data, which he/she needs, and when needed. But still the process of culling out useful information from database is in the purview of the designers, programmers or managers who are well versed with database specific query languages. These languages scare away naive end users who are left at the mercy of the programmers who provide them with limited predefined queries to extract information from the database. But the scope of these queries is generally restricted to routine information and any new queries have to be designed by programmers and handed over to end-users. Moreover the end users are not at all aware of the underlying schema, which can help them design queries if at all they know some querying language. As data storage techniques improved over the years organizations rapidly opted for the change and not only collected large volumes of data but in different sources and formats eg .txt, .xml, .dbf, .html etc

Querying relational databases requires users to be aware of the underlying database schema and also the knowledge of the structured query language specific to the database. Most of the business database needs lots of reports to be derived/extracted from the underlying data for analysis and decision making purposes. This is accomplished by the use of pre-designed formats that accepts specific input and generates output in the specified format. Problem does not end here, data is not only stored in databases but also in different file formats eg .txt, .html, .xml etc were data retrieval rules are all together different and vary from format to format. Retrieving data from files requires users to be aware of the underlying file schema and also the knowledge of the retrieval method used, specific to the file format.

User is also required to have data storage knowledge, as to where data of his/her interest is stored. It becomes very complicated as there can be n files and n tables. User is then either supposed to depend on programmer or memories where data of his interest is stored, thing get over complicated because of data replication- same data can be stored in n file and n table at the same time. User does not want to depend on programmer neither have patience to memories where data is stored, user expects to give query[4] without specifying where data is stored and in which format it is stored

It's not only about data retrieval but also about data translation. Data retrieved from databases are all together in different form as compared to data retrieved form files eg .txt, .xml etc. Users are interested in having data presented in specific form and not according to their storage pattern, in other words data stored in database will be say in columnar form were as data stored in .txt file will be all together in different form as compared to .html file, irrespective of data storage pattern and data retrieval technique data has to be presented in generic format, where user can decide the format himself/herself and data presentation hides data storage format and data retrieval technique[17]

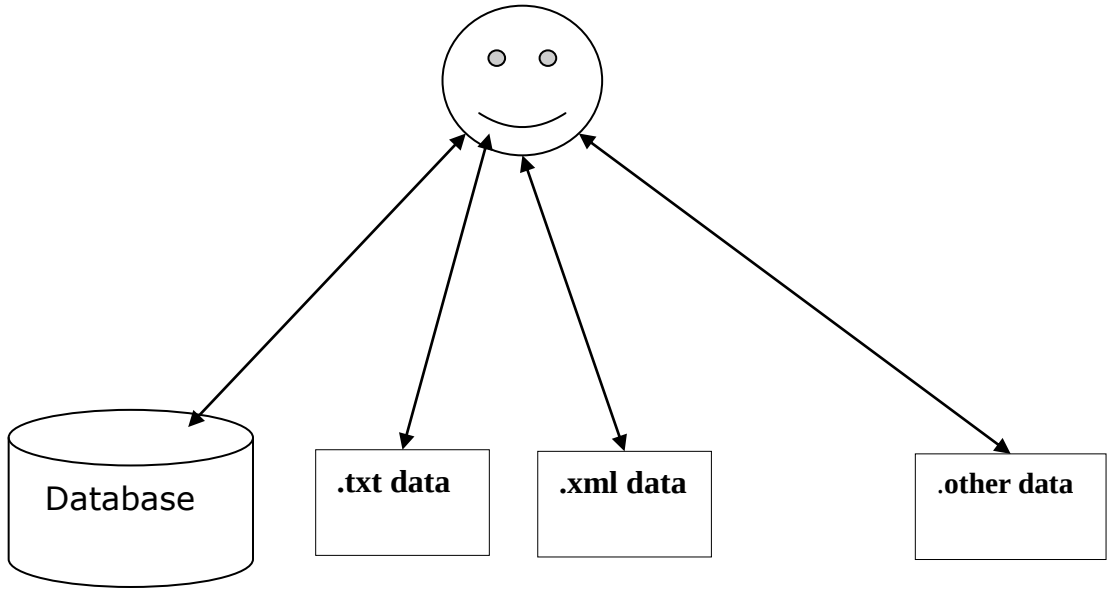

Figure 1: Generic Scenario

2. DATA WAREHOUSE & DATA MINING

Data Warehouse was need of the hour in enterprises in order to analyze business process for better decision making as per founder of Data Warehouse Inmon: "A Data Warehouse is a subject oriented, integrated, non-volatile and time-variant collection of data in support of management's decisions".

Data Mining is the process of extracting/exploring data from n data sources and re-organizing it for purposes other than what the databases were originally intended for. What data is to be mined varies from company to company, user to user in other words depends on the nature and organization of the data, so there can be no such thing as a generic "data mining tool".Of course the data must be continuously refreshed, so the scrubbing and reconciliation process must be a permanent feature of the Warehouse, and will have to be modified every time the databases are modified or new databases become available.

Creating and maintaining a Data Warehouse is a huge job even for the largest companies. It can take a long time and cost a lot of money. In fact, it is such a major project companies are turning to Data Mart solutions instead.A Data Mart is an index and extraction system[2]. Rather than bring all the company's data into a single warehouse, the data mart knows what data each database contains and how to extract information from multiple databases when asked17].

Creating a Data Mart can be considered the "quick and dirty" solution, because the data from different databases is not scrubbed and reconciled, but it may be the difference between having information available and not having it available[17].

OLAP can be used for data mining or the discovery of previously undiscerned relationships between data items. An OLAP database does not need to be as large as a data warehouse, since not all transactional data is needed for trend analysis. Using Open Database Connectivity (ODBC), data can be imported from existing relational databases to create a multidimensional database for OLAP [18].

3. ONLINE ANALYTICAL PROCESSING AND DATA MINING

Development of Keyword preprocessor

Preprocessor is meant to check, correct, rearrange/modify user input as and when required. It not only will check & correct spelling mistakes if any making use of traditional dictionary, but also delete extra space, unnecessary ",.;" and if required rearrange the words, making the best arrangement of user input without damaging the content and context of the input. Preprocessor will have access to organization main databases like employees thus enabling it to access such database to make user input to more precise queries[7][13][14].

Construction of knowledge base

Defining how our knowledge base will be set up is a crucial first step before we initially populate our knowledge base. If our knowledge base is not well organized and actively managed, the answers can easily become outdated and the information can become disjointed. As a result, finding answers can become more difficult for our users. In addition to initial planning, we need to develop specific procedures for maintaining the knowledge base over the long term so that we can keep the information organized and updated to maximize the effectiveness of our knowledge base.

Before we initially populate our knowledge base with question/answer pairs, we must plan for the growth of the knowledge base by defining and organizing the information you want to present. By organizing information into distinct and logical categories, the information is more accessible and will avoid having to reorganize our knowledge base as it becomes larger. Once our class, categories, and custom fields are in place, we can then develop the processes for proposing, publishing, reviewing & if required executing queries[7].

To design and build an effective knowledge base, and

ultimately to manage it effectively, we start by addressing the following areas:

a. Consider the amount of information to be presented
b. Identify our audience and the scope of information to be included
c. Define users and categories
d. Define additional custom fields as necessary
e. Develop writing and style guidelines
f. Designate responsibilities for managing our knowledge base
g. Define a process for proposing new answers
h. Define an approval review process for new answers
i. Determine the display position of new answers on the answers lists
j. Notify users of new answers, if any
k. Evaluate customer feedback
l. Determine a process for reviewing existing answers

Creating Mediator Based Integration

Mediator based integration architectures [6][9][11] define a framework to deal with such problems. In such systems, the mediator holds a schema (mediator schema) which semantically subsumes the interesting parts of the source schemas. Technical and syntactical heterogeneity in the sources is hidden by wrappers which offer a uniform interface to the mediator. In our approach this interface is comprised of a relational export schema (source schema) and the set of possible queries[7] against this schema. The mediator tries to find answers for queries against the mediator schema by combining data from different sources which are accessed through their wrappers. In this process, many types of schematic and semantic discrepancies have to be bridged.

Primary and Secondary Classifiers

Primary and Secondary Classifiers play a major role in the Data Mining process. The algorithm involved is shown as under:

Step 1: Get Input Query from the User.

Step 2: Perform Preprocessing like checking syntax, size and data type involvement.

Step 3: Feed the query to a Primary Classifier. If already such kind of information is mined in the system then using Knowledge base and Data Mart provide the generic information to the user. If there is a failure go to step 4.

Step 4: Broadcast this query to various Heterogeneous data sources[3][10][15][16]. Also update the user profile information data source and transfer the output to mediator data source.

Step 5: The large data volume present in the mediator data Source is provided to the secondary classifier.

Step 6: Using a Knowledge base information and Mediator Data Base information proper information is filtered. If there is a success then corresponding entries are made in the Knowledge base and the resultant data is passed to the user via a data mart.

Step 7: If there is a failure go back to step 4 till generic information is not mined.

Step 8: Clear the Mediator Temp Data Source for next Query.

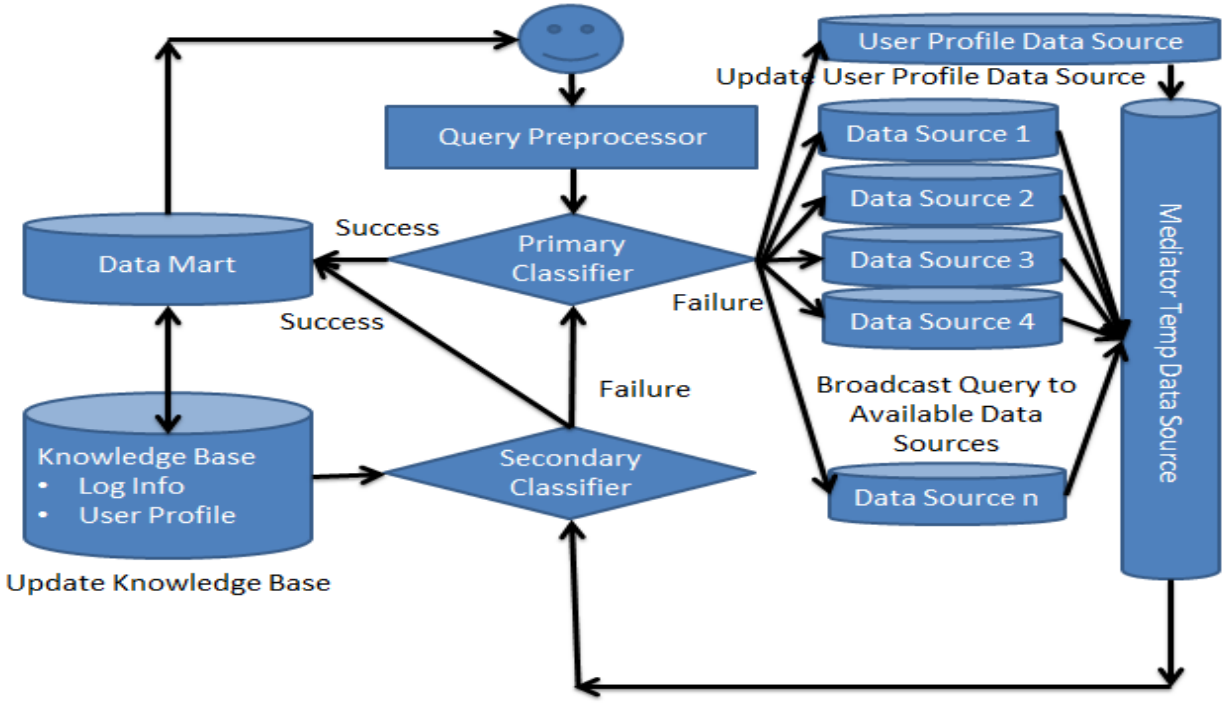

Figure 2: Generic Search Optimization

4. PERFORMANCE DETAILS

The performance details for the system under investigation is shown in the graph shown below. A set of queries were made to run on the system and it was observed that after some training trails the search time was reduced to a large extent and after that remained constant. This was due to direct access from the data mart as the system got trained to such queries as the knowledge and user profile data source base were updated concurrently thus reducing the search time.

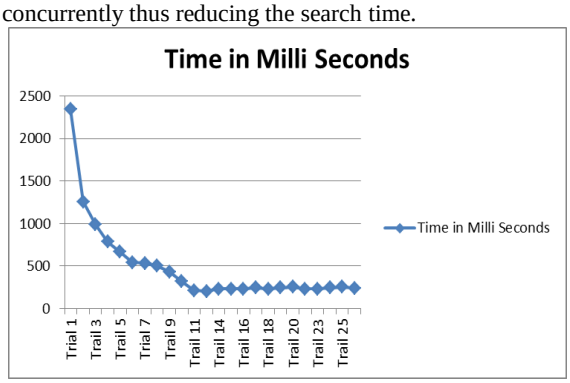

Figure 3: Performance Graph for Generic Search

5. CONCLUSION

Data retrieval across multiple heterogeneous[15][16] data sources is still a challenge at large because of many reasons including m formats, requires users to be aware of the underlying database schema and also the knowledge of the structured query language specific to the database, data inconsistency across n sources, most of the enterprise database needs lots of reports to be derived/extracted from the underlying data for analysis and decision making purposes & these reports can neither be generic and nor can be determined in advance.

To overcome such problems we proposed GENERIC SEARCH PRINCPLE where in users of the organization irrespective of their technical ability, data source knowledge and location can search n heterogeneous data sources including legacy data sources of organization and retrieve consistent information. This principle also taking into consideration user attributes like his/her location, work profile, designation etc so as to make search more relevant and results more precise.

6. REFERENCES

[1] R. Ashok Kumar, Dr Y. Rama Devi, "Efficient Approaches for Record level Web Information Extraction Systems". Published in International Journal of Advanced Engineering & Application, pp 161-164, Jan 2011

[2] Tari, L. Tu, P. Hakenberg, J. Chen, Y. Son, T. Gonzalez, G. Baral, "Incremental Information Extraction Using Relational Databases". Knowledge and Data Engineering, IEEE Transactions on Issue:99 , pp 25-35, 28 October 2010

[3] Mohammad Ghulam Ali, "Object Oriented Approach for integration of heterogeneous databases in a multidatabase system and local schemas modifications propagation", international journal of computer sciences and information security, vol 6, No. 2, 2009

[4] J. Huang and E. Efthimiadis, "Analyzing and evaluating query reformulation strategies in web search logs". In Proceedings of CIKM, pp 77-86, ACM, 2009.

[5] Ramakrishna Srikant, Sugato Basu, Ni Wang, Daryl Pregibon, "User browsing models: relevance versus examination". In Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 223-232, 2010.

[6] Md. Sumon Shahriar and Jixue Liu, "Constraint-Based Data Transformation for Integration: An Information System Approach", International Journal of Database Theory and Application Vol. 3, No. 1,pp 85-92, March, 2010.

[7] E. Alfonseca, K. Hall, and S. Hartmann, "Large-scale computation of distributional similarities for queries". In Proceedings of NAACL-HLT, Association for Computational Linguistics, pp 29-32,2009.

[8] Bo Yang and Manohar Mareboyana, "Progressive Content-Sensitive Data Retrieval in Sensor Networks". Journal of Computer Science 5 (7):pp 529-535, 2009.

[9] Stefan Biffl, Wikan Danar Sunindyo, Thomas Moser, "Semantic Integration of Heterogeneous Data Sources for Monitoring Frequent-Release Software Projects". International Conference on Complex, Intelligent and Software Intensive Systems, 2010.

[10] Marc Van Cappellen, Wouter Cordewiner, Carlo Innocenti, "Data Aggregation, Heterogeneous Data Sources and Streaming Processing: How Can XQuery Help? Bulletin of the IEEE Computer Society, Technical Committee on Data Engineering, 2008.

[11] Alon Halevy, "Information Integration". In Encyclopedia of Database Systems, 2009.

[12] Peter Pach, Attila Gyenesei, and Janos Abonyi, "Compact fuzzy association rule based classifier". Expert Systems with Applications, 2007.

[13] S. Agarwal, S. Chaudhary, and G. Das. 'Dbxplorer, "A system for keyword based search over Relational Databases". In proceedings of ICDE 2002.

[14] N.L. Sarda & Ankur Jain. "A System for Keyword-based Searching in Databases."

[15] Srujana Merugu & Joydeep Ghosh "A Distributed Learning Framework for Heterogeneous Data Sources". KDD'05, August 21–24, 2005, Chicago, Illinois, USA.

[16] Ulf Leser. "Combining Heterogeneous Data Sources through Query Correspondence Assertions".

[17] Automation Access: http://www.aaxnet.com

[18] Search Data Management http://searchdatamanagement.techtarget.com