WordNet Ontology Based Query Reformulation and Optimization using Disjunctive Clause Elimination

[1]K.Amshakala
Department of Computer Science and Technology
Coimbatore Institute of Technology, Coimbatore,India
Email:amshakala@cit.edu.in
Mobile: 9095260060

[2]Dr. R.Nedunchezhian
Professor and Head, Department of Information Technology
Sri Ramakrishna Engineering College, Coimbatore, India
Email:rajuchezhian@gmail.com
Mobile:9842523005

Abstract:

Data integration systems attempt to provide users with seamless and flexible access to information from multiple autonomous, distributed and heterogeneous data sources through a unified query interface. Besides data are continuously growing, maintained by different organizations and managed autonomously, querying data from heterogeneous data sources faces new challenges. As data integration has been automated, the ambiguity in concept interpretation also known as semantic heterogeneity has become one of the main obstacles to this process. Introduction of the Semantic Web Vision [5] and the shift towards machine-understandable web resources have underscored the importance of automatic semantic integration of data elements. Ontologies[7] have been widely accepted as the model of choice for modeling heterogeneous data sources by various communities including the areas of databases, Knowledge representation and Information retrieval. WordNet ontology[3] is a large lexical database that is used in many schema matching algorithms to match schemas based on the semantics of attributes. In this paper ontology based semantic query reformulation technique is followed to improve the recall of the query. The reformulated query is optimized by removing disjunctive clauses in the query to reduce the computational cost of the semantic query execution. Experimental results show that the proposed optimization technique improves recall with minimal execution time.

Keywords:

Data integration, Semantic Web, Ontology, Semantic heterogeneity, Semantic queries, Semantic query reformulation

DOI: 10.5121/ijdms.2011.3405

1. INTRODUCTION

Integrating data from distributed, heterogeneous, autonomous data sources is a fundamental requirement in many database application domains, such as cooperative information systems, e-business, data warehousing, and semantic query processing. Ideally, a data integration system should allow users to focus on what information is needed without having to offer detailed instructions on how to obtain the information[10,11,14]. Thus, in general, a data integration system must provide mechanisms for communications and interaction with each data source as needed, Specification of a query, expressed in terms of a common vocabulary, across multiple heterogeneous and autonomous data sources, transformation of such a query into a plan for extracting the needed information by interacting with the relevant data sources, and combination and presentation of the results in terms of a vocabulary known to the user. Thus, bridging the syntactic and semantic gaps among the individual data sources and the user is a key problem in data integration[14]. To date, all approaches to wrap data collections, or even to create mappings across comparable datasets, require manual effort. Despite some promising work, the automated creation of such mappings is still in its infancy, since equivalences and differences manifest themselves at all levels, from individual data values through metadata to the explanatory text surrounding the data collection as a whole.

There are different kinds of heterogeneity like syntactic, schematic and semantic heterogeneities existing among the data sources that are to be integrated. *syntactic heterogeneity*, is caused by different languages used for modeling the different sources, *schematic heterogeneity*, results from different structures of source schemas, and *semantic heterogeneity*, arises when different sources contain instances with different meanings or interpretations[14]. Richer semantics on structure and data vales of data sources are required for handling these heterogeneities. Currently, research work on the Semantic Web[5] and data integration[14] are focusing on using *ontologies which* is a formal and explicit specification of a shared conceptualization [7] as semantic support for data processing. The use of ontologies can benefit data integration tasks in a variety of ways, including metadata representation, global conceptualization, support for high-level queries, declarative mediation, and mapping support. Ontologies have proven to be useful to capture the semantic content of data sources and to unify the semantic relationships between heterogeneous structures[1,2,8,15]. Thus, users should not care about where and how the data are organized in the sources. In this paper our focus is on query reformulation in an optimized way , with the support of WordNet ontology[3] to overcome semantic heterogeneity.

2. RELATED WORK

Use of the semantic knowledge in query generation and query reformulation is used in two ways in database management systems, one for query optimization and other for data integration[1,10]. Semantic queries are expanded either manually or automatically or interactively. The work on conceptual query expansion for document retrieval is presented in [16]. They use terms of initial result of the user's query as candidates for the query expansion and limit the candidate terms using formal concept analysis. Some work has been done in [6,5,8,13,15] to improve the recall of semantic queries for a single database, but still there is information loss due to unhandling of some natural language relationships like acronym and lexical variants. First well founded intelligent query interface for formulating queries is presented in [6]. Context based query evaluation is another technique used to improve recall in information retrieval systems [17]. Context at system level, query level and user level were exploited in [17] to improve the recall of

the returned results. Semantic query expansion is addressed in [1,2,8,12,13] as a semantic problem rather than as a pattern matching problem. As a consequence, if we consider semantics in query processing, the number of results for a submitted query substantially increases. Compared to query optimization methods this approach is not to speed up query processing but to provide users with additional meaningful answers. To this end, a set of reformulation rules and practical examples were presented to show their usefulness in [1,2,8]. In [1], a new approach on how to improve the answers of queries based on semantic knowledge expressed in ontologies is presented. In addition to reformulation rules proposed in [2,8], this work includes "Composite rule" which considers part-whole relationship between objects. [18] has addressed query routing, schema matching and query optimization problems in P2P data sharing systems where they have used domain ontology to provide semantic knowledge on the data sources.

In this paper we have described an automated source specific query reformulation technique with the support of ontology. Here our aim of the query reformulation in data integration is to provide transparent access and extend the results of a given query in a semantically meaningful way which best captures the user's information needs, even if he/she does not have knowledge of the domain, internal structure of the data sources and the contents of the data sources. The reformulated query is also optimized by decomposing it into source specific queries and hence the cost of query execution is not compromised for improving accuracy of query results.

3. PROBLEM DEFINITION

Vocabulary Enhancement Rule proposed in [1] considers SynOf (Synonym) and IsA (Specialization) relationships to expand the user submitted queries. For example , table 1 shows a product table stored at a single data source.

Product_ID	Product_Name	Price
1	computer	30000
2	PC	25000
3	IntelPC	50000
4	Monitor	5000
5	Keyboard	300
6	Data processor	40000
7	Calculator	5000

Table 1: PRODUCT table

A simple user query over PRODUCT table on the product "computer" is formulated as follows

Q_1= Select price from PRODUCT where Product_Name = "computer";

On investigating the product ontology it is found that "data processor" and "PC" are synonym to the "computer", so they must also be queried. The query on "PC" is expanded as follows.

Q_1' = Select price from PRODUCT where Product_Name = "computer" OR Product_Name = " PC" OR Product_Name = "Data processor";

The query Q_1' yields a better recall compared to query Q_1 since it retrieves records that are semantically related to the query term. Semantic expansion of user queries will yield better results

in centralized database systems that may have different records describing the same entity. When the same entity is recorded using different names in the database, it is said to contain redundant information. In practice, the scenario described above occurs very rarely. So semantically enriching the user query by adding synonyms of the query terms using disjunctive clause is an unnecessary effort. In distributed autonomous database systems, the same entity will be stored in different names in different data sources. In such cases, semantic query expansion will not improve the recall of query results, since at a single data source each entity will be called by a unique name. Semantic query decomposition approach will retrieve the same set of results from all the data sources by submitting source specific queries at each data source instead of semantic query expansion.

Semantically expanded queries are computationally costlier because of the disjunctive OR clauses connecting the synonyms of query term. These disjunctive clauses are much harder to process and optimize because of the following reasons. With queries including disjunctive clause, only little optimization can be done because the rows satisfying the query are the union of the rows satisfying each of the individual conditions. If any one of the search conditions does not have an access path, then the query optimizer is compelled to choose a full table scan to satisfy the query. Performance can only be improved if an access path exists on every condition in the disjunctive clause[4]. In this case, row sets can be found satisfying each condition and then combined through applying a union operation across the result sets to eliminate duplicate rows. However, set union operations can also be expensive. The customary way to implement union operations is to sort the relations on the same attributes and then scan the sorted files to eliminate duplicate rows. In many cases the use of disjunctive clauses in queries results in either a brute force linear search of the table, or a sort of a potentially large amount of data.

Queries formulated using an SQL query language provide little predictive information useful for estimating query performance. Internal knowledge of the database structure, data distribution, and query optimizing strategy are necessary to develop effective query statements. This technical knowledge rarely exists in the user community. This leads us to recommend that enterprise decision support systems remain independent from user developed, unstructured queries. Any request to integrate ineffective or unproven query statements into a management system should be discouraged. The inevitable result is a dissatisfied client[4]. It is very difficult for naïve users to retrieve desired data from heterogeneous, autonomous and geographically distributed data sources because of their lack of knowledge about the internal structures and contents of data sources. In such cases, ontological support is required to form source specific queries. In the proposed approach, source specific semantic query reformulation technique is implemented with the Wordnet ontology[3] support.

4. SEMATIC QUERY REFORMULATION.

Semantic query reformulation is used to answer user's queries in more appropriate manner by reformulating user submitted queries into semantically enriched queries. These semantically enriched queries are called as semantic queries[1,2,8]. The semantic queries are formed by using ontological support, which provide terms semantically equivalent to query term. Query reformulation approach also called semantic query optimization, takes advantage of the semantic knowledge about the contents of databases for optimization. The basic idea is to use the knowledge to reformulate a query into a less expensive yet equivalent query. The main goal of semantic query reformulation proposed in this paper is to translate a user query into a set of

queries which best suite the structure of the distributed sources. This reformulated query reduces loss of information during data integration and extends the result of a given query in a semantically meaningful way.

The architecture of the semantic query reformulation system shown in figure1 is based on the mediator architecture described in [9]. It is a three layered architecture which introduces a central mediation layer that separates user-oriented processing and database access. A mediator is a software module that exploits encoded knowledge about some sets or subsets of data to create information for higher level of applications [9]. A mediator in a query processing system could be used to access multiple data sources. The user submits a query defined in his terms through the user interface, which will be converted by the query generator into an appropriate SQL query. The mediator includes a query interface, query generator, query reformulation module, ontology reasoner and result presentation module.

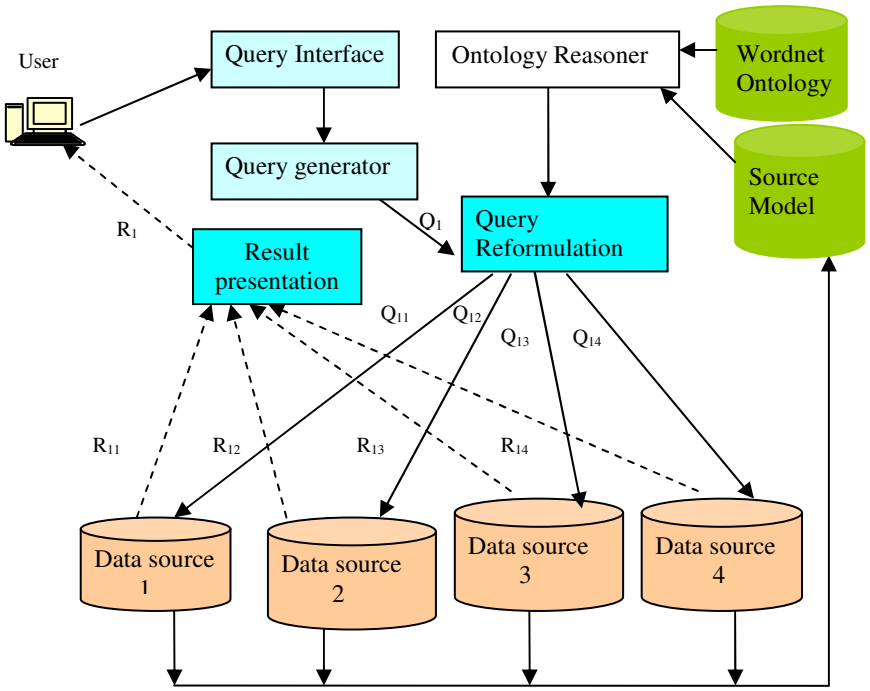

Figure 1: Query processing architecture

A source model is build by extracting metadata from the distributed data sources. For query reformulation, the proposed approach relies on additional synonyms extracted from ontology. Wordnet ontology is used to generate semantically meaningful queries by deriving the synonyms related to the query terms. The source model is extended by adding distinct values of a particular query attribute from all the relational databases. The Ontology Reasoner is responsible for grouping the distinct domain values based on their synonyms and constructs a domain-value table which has a structure as shown in table 2. The number of rows in the table is equal to the number of distinct values available in the distributed databases. Each row in the domain table represents a unique domain value. Each column in the table represents different data sources. Every cell $_{ij}$ in the table contains the synonym of the domain value i present in the source database j. If a source does not contain a synonym corresponding to a domain value, the cell is filled with null value.

The construction of source model is only once and updated whenever there are changes in the distributed data sources.

	Source DB1	Source DB2	Source DB3	Source DB4
Domain value 1	Syn_{11}	Syn_{12}	Syn_{13}	Syn_{14}
Domain value 2	Syn_{21}	Syn_{22}	Syn_{23}	Syn_{24}
Domain Value 3	Syn_{31}	Syn_{32}	Syn_{33}	null
Domain value 4	null	Syn_{42}	Syn_{43}	Syn_{44}

Table2: Domain-value Table

For example, table 4 shows a domain-value table constructed for distributed databases shown in table 3 (a- d).

ID	Product_Name	Price
1	mobile phone	4000
2	TV	25000
3	Stereo	5000
4	PC	20000
5	Washer	10000

Table 3(a)- PRODUCT

ID	Product_Name	Price
1	cellular phone	5000
2	Television	20000
3	Sound system	5000
4	Personal computer	30000

Table 3(b)- ITEM

ID	Product_Name	Price
1	mobile	3000
2	Telecast	25000
3	Music system	4000
4	Computer	25000
5	Washing machine	8000

Table3(c)-COMMODITY

ID	Product_Name	Price
1	Music system	3000
2	Monitor	10000
3	Keyboard	2000
4	Calculator	1000

Table3(d)-GOODS

	PRODUCT	ITEM	COMMO-DITY	GOODS
Mobile phone	mobile phone	cellular phone	mobile	cell
TV	TV	Television	Telecast	null
Music system	Stereo	Sound system	Music system	Music system
Computer	PC	Personal computer	Computer	null
Washing machine	Washer	null	Washing machine	null

Table 4: Domain-value Table for distributed databases of Table 3(a-d)

The constructed domain-value table is also added to the source model in order to use it for future query processing. The Query reformulation module uses the Domain-value table to reformulate the user submitted query by replacing the query condition with the source specific domain value which could be one of the synonyms of the query term and generates source specific queries. The reformulated queries could be distributed to appropriate data sources to retrieve the results.

For example, the query term in query Q_1 is "computer". From domain-value table, the synonyms of the query term stored at each source are retrieved by accessing the corresponding row. The database table names are retrieved from row 1 in the table. In [3], the query Q_1 is simply expanded by adding the synonyms of the query term using disjunctive clauses to reduce information loss as shown by query Q_1'. The query Q_1, instead can be reformulated into source specific queries as follows.

Q_{11} = Select price from PRODUCT where Product_Name = " PC";
Q_{12} = Select price from ITEM where Product_Name = "personal computer";
Q_{13} = Select price from COMODITY where Product_Name = "computer";
Q_{14} = Select price from PRODUCT where Product_Name = "Data processor";

The reformulated queries are sent to corresponding data sources and finally the results are combined from all the data sources. The decomposed queries Q_{11}, Q_{12}, Q_{13}, Q_{14} does not include disjunctive clauses and hence the query execution cost is reduced. Unlike [3] where query Q_1 is semantically expanded, the proposed approach follows semantic decomposition of the semantically enriched query Q_1'. The information loss in integration of several data sources is reduced and at the same time the execution cost is also reduced. The results returned by the data sources will be presented to the user in the user required form and this is the responsibility of the result presentation module.

The semantic query reformulation technique proposed in this paper increases the recall of the query with minimized execution cost. This is proved by carrying out an experimental evaluation of the technique and the results are discussed in the following section.

5. EXPERIMENTAL RESULTS

In this section we present the evaluation of our proposed approach with respect to technique of [3]. In order to make comparison of the results, we have created test bed using Processor 3.0GHz, RAM 1GB and Windows XP on the standalone PC. The dataset for the experiments are synthesized by duplicating the records of the databases shown in table 3(a-d). On this test bed we executed queries for both techniques and analyzed the results.

The generation of source specific queries as proposed in this paper reduces the execution time of the user submitted queries compared to the query expansion rule proposed in [1]. The removal of disjunctive clause OR has brought down the execution time to a considerable amount and hence the results are presented to the users promptly. As the graph shows in Figure 2, the proposed optimized query reformulation method reduces the execution time by 50% compared to that of the un-optimized query expansion method proposed in [1] .

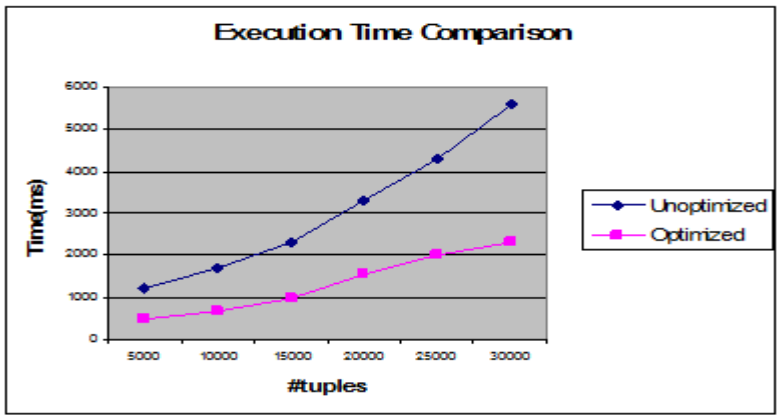

Figure 2: Execution Time (Query expansion Vs Query Decomposition)

6. CONCLUSION

In this paper**,** an ontology based query reformulation technique is proposed in order to query distributed autonomous data bases. The query reformulation is done by using a source model containing metadata of the underlying distributed data sources and also domain specific information. Semantic query reformulation is done in a source specific manner without blindly expanding query terms. By following semantic query decomposition instead of query expansion, the execution time of data retrieval is reduced without compromising on the number of relevant records retrieved. In future this work could be improved by applying techniques to construct domain specific information on the fly to meet the growing data sharing needs.

REFERENCES

[1] Waris Ali , Sharifullah Khan, “Ontology Driven Query Expansion in Data Integration”, Fourth International IEEE Conference on Semantics, Knowledge and Grid , 2005, pp 57-63.

[2] Chokri Ben Necib and Johann-Christoph Freytag, "Using Ontologies for Database Query Reformulation, Advances in Databases and Information Systems" (ADBIS), Hungary, 2004.

[3] Wordnet Documentation : http://wordnet.princeton.edu.

[4] William miles , "Optimizing SQL Query Processing" , InterviewInfo.net 2005.

[5] Tim Berness Lee, "Semantic web", The Scientific American,2001.

[6] D. Bonino, F. Corno, L. Farinetti, and A. Bosca, "Ontology Driven Semantic Search", World Scientific and Engineering Academy and Society (WSEAS) Transaction on Information Science and Application, Vol. 1, pp. 1597-1605, 2004.

[7] Gruber T., "A translation approach to portable ontology specifications", Journal of Knowledge Acquisition, Vol. 2, USA, pp 199-220, 1993.

[8] Necib, C.B., Freytag, J., "Ontology based Query Processing in Database Management Systems" , In the Proceeding of the 6th international on Ontologies Databases and Application Semantics (ODBASE), pp. 37-99, 2003.

[9] G. Wiederhold, "Mediators in the architecture for future Information Systems", IEEE Computer Magazine, pp. 38-49, 1992.

[10] Sharifullah Khan and Franck Morvan, "Integrating Biomedical Sources on the Internet", In the proceedings of ISCA 19th International Conference on Parallel and Distributed Computing Systems (PDCS), pp. 165-170, USA, 2006.

[11] C. A. Globe, and N. W. Paton, "Transparent access to multiple bioinformatics information sources", IBM System Journal, pp. 532-551, 2001.

[12] Williams, and Martha E., "Query Expansion, Annual Review of Information Systems and Technology" (ARIST), http://faculty.washington.edu/efthimis/pubs/Pubs/qe-arist/QE-arist.html, pp 121-187, 1996.

[13] Steve Renals, "Query Expansion",http://homepages.inf.ed.ac.uk/srenals/ pubs/1999/esca99-thisl/node6.html, 1999.

[14] Maurizio Lenzerini, "Data Integration: A Theoretical Perspective", Proceedings of the twenty-first ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems, pp 233-246, 2002.

[15] Agustina Buccella, Alejandra Cechich, Nieves R. Brisaboa , "An Ontology Approach to Data Integration" , Journal of computer science and technology,2003.

[16] F.A. Grootjen and Th. P. van der Weide, "Conceptual Query Expansion", Journal of Data Knowledge and Engineering, pp. 174-193, 2004.

[17] Abdelkrim Bouramoul, Mohamed-Khireddine Kholladi and Bich-Lien Doan, "Using Context to Improve the Evaluation of Information Retrieval Systems",International Journal of Database Management Systems (IJDMS), Vol.3, No.2, May 2011.

[18] Raddad Al King, Abdelkader Hameurlain, Franck Morvan, "Query Routing and Processing in Peer-to-Peer Data Sharing Systems", International Journal of Database Management Systems (IJDMS), Vol.2, No.2, May 2010.