NATURAL LANGUAGE QUERY PROCESSING BASED ON FUZZIFICATION TECHNIQUE

Dr. ASHISH KUMAR TAMRAKAR[1]
Dr. VISHNU KUMAR MISHRA[2]
Dr. MEGHA MISHRA[3]

[1]Assistant Professor, RCST, DURG, Chhattisgarh, India

[2]Associate Professor, SSGI-SSTC, BHILAI, Chhattisgarh, India

[3]Associate Professor, SSGI-SSTC, BHILAI, Chhattisgarh, India

ABSTRACT

The modern information based systems to make them more human friendly. Dealing with information inaccuracy, fuzzy techniques have been widely integrated with different database models and theories. The traditional query in relational database is unable to satisfy the needs for dealing with fuzzy linguistic values. In this paper, a new data query technique composed of fuzzy theory and MS-SQL is provided. Here, the query can be implemented for fuzzy linguistic variables query via an interface to JAVA SWING. Natural language processing based systems are extremely capable to represent and manipulate the complex query as complicated and uncertain relationship existing among them. They are also much appropriate for engineering and scientific applications, managing with large data intensive applications. In this paper, a survey of different approaches regarding integration of fuzzy techniques in natural language complex query processing has been discussed, related to numerous types of conceptual data modeling, querying and indexing. Now I will try to build a software tool which can perform the work on complex query and we can fetch the information from the database very easily.

KEYWORDS: A Natural Language, Fuzzy Sets and Fuzzy Logic, Fuzzy Query, Fuzzy Theory, MS-SQL, Relational Database.

INTRODUCTION

Complex structure query is useful due to its lots of application area in human communication language while Structural Query Language (SQL) is very restrictive in data extraction. So complex query processing may easily controlled using natural language processing.

Database management systems (DBMS) are widely used software products in many kinds of systems. As we know natural language is the main communication method for humans, but this causes it difficult to handle imperfect information.

In order to obtain query results, there are two basic methods of research in the use of SQL Combined fuzzy theory in DBMS. The first is still to build a classic relation database, only to modify or extend SQL query by transforming query conditions to a fuzzy scope. After that, change it to precise SQL clause. This procedure is easy and consistent with ordinary query, but lacks flexibility. Sometime it is apt to produce query errors. The second approach is to assume that the database is fuzzy sets and fuzzy logic is used to make it easier and more human consistent. This is mainly done by constructing a database model based on fuzzy logic.

Natural language processing is advance form of artificial intelligence where human understandable natural language is used to create communication. It helps for taking decision and getting result by solving and computing queries.

Query optimization is the process of producing an optimal (close to optimal) query execution plan which represents an execution strategy for the query. The main task in query optimization is to consider different orderings of the operations and minimize total cost associated with execution of request.

Classical SQL queries have remarkable capabilities in terms of data extraction and answer formation from information stored at widely dispersed databases. Human queries are rarely crisp which poses challenges in efficient answer formation and data retrieval. These are based on human perception which is grossly inexact and imprecise based on world knowledge. Integration of query languages with fuzzy logic can increase their capability in data retrieval based on human perception. Query optimization is a crucial and difficult part of the overall query processing. This paper deals with the second method. At beginning, consider a conventional patients database. Then, consider the second method to cover the fuzzy query in patient's database. Third, present how to define fuzzy elements appearing in the query, mainly the transformation of a fuzzy query into its equivalent legitimate MS- SQL query.

RELATED WORK

Join queries in Complex query based database are ship whole, fetch as needed, semi joins and bloom joins and each of these join strategies are having their own advantages and disadvantages[1].

Main considerations of query processing in Complex query based database databases are: communication cost, if there is several copies of a relation, decide which copy to use, amount of data being shipped, relative processing speed at each site and site selection.

In a relational database [2] all information can be found in a series of tables. A query therefore consists of operations on tables. The most common queries are Select-Project-Join queries. Complex query processing transforms a high-level query (of relational calculus/SQL) on a Complex query based database (i.e., a set of global relations) into an equivalent and efficient lower-level query (of relational algebra) on relation fragments. Complex query processing is more complex because of fragmentation/replication of relations, additional communication costs and parallel execution of operations.

The very first attempts at NLP database interfaces are just as old as any other NLP research. In fact database NLP may be one of the most important successes in NLP since it began. Asking questions to databases in natural language is a very convenient and easy method of data access, especially for casual users who do not understand complicated database query languages such as SQL [3].

The main purpose of Natural Language Query Processing is to interpret an English sentence and hence a complementary action is taken. Querying to databases in natural language is a convenient method for data access, especially for newbie's who have less knowledge about complicated database query languages such as SQL.

METHODOLOGY

The expected scheme is implemented using below scheme-

- Create a different partitions for the database (here three database are created with different schemas).
- Apply workload i.e. queries on all databases and estimate the cost of each query in terms of response time.
- The number of queries applied on the partitions is varied from 1 to 100.
- Different types of query operations such as Insert, Update, delete, fetch and fuzzy value based searching are applied on all the partitions
- Fuzzy range is defined for the salary attributed defined in all the partitions.
- If user is fetching records of particular fuzzy ranges then query is classified into fuzzy range1, 2 or 3 as per the user input.

- Then the comparison is done on the basis of cost of queries i.e. response time of the query.

FUZZY APPROACHES

This approach has been used to extend database systems in storing and updating imprecise information (data) and in processing imprecise queries. Consider a fuzzy query: find name, grade of quite good students and just tall students where age > 15. This query includes two fuzzy concepts: good student and tall student and one crisp query criteria (i.e. age > 15). The integration of a query processing subsystem into a Complex query based database management system with fuzzy logic is used for analyzing query response time across fragmentations of global relations.

Query interpreter [5] works as basic interaction platform between user and the rest of database model. Input to query interpreter is a formal fuzzy query through some query interface or query file (depending on the database system being used) and output is a logical fuzzy query. So, query interpreter transforms the formal fuzzy query into logical fuzzy query.

It has been presented that a fuzzy query interpreter which transforms fuzzy queries to the classical SQL structure and queries based on linguistic[4] expressions on client side are supported. Many others fuzzy query interpreter in their approaches towards fuzzy query processing.

The remarkable ability of fuzzy inference [6] engines in making reasonable decisions in an environment of imprecision and uncertainty makes them particularly suitable for applications involving risks, uncertainty and ambiguity that require flexibility and tolerance to imprecise values. In a relational database all information can be found in series of tables. A query therefore consists of operations on tables. The most common queries are Select-Project-Join queries. For a given query, the search space can be defined as the set of equivalent operator trees that can be produced using transformation rules. Complex query optimization process consists of transforming global queries from control sites to Fragment queries in local site. The process of transformation is as shown below:

Table 1: Comparison of Time Complexity for Various Relational Operations

Operations	Time Complexity
Select, Project (without duplicate elimination)	O(n)
Project (with duplicate elimination)	O(n log n)
Group Join Semi-join Division Set Operators	O(n log n)
Cartesian Product	O(n2)

The main feature of fuzzy logic is that it is able to deal with imprecise linguistic information which makes it attractive for automatic text summarization from the mathematical viewpoint; complex-valued fuzzy sets are natural. In classical (2-valued) logic, every statement is either true or false. In the computer, “true” is usually represented as 1, and “false” as 0. As a result, in the 2-valued logic, the set of possible truth values is a 2-element set . The traditional 2-valued logic is well equipped to represent:

- Situations when we are completely sure that a given Statement is true and
- Situations when we are completely sure that a given Statement is false.

An important issue which is re-emerging in the field of relational database management systems is the ability for non-expert users to access stored data using the more powerful aspects of the structured query language (SQL).

The widespread use of relational database management systems in industry as well in scientific research has increased the need for a solution to this issue. A proposed method to allow non-expert users more successfully obtain data is the use of an artificial intelligence application to process natural language from the user in the

Form of a question or sentence into a SQL statement. This paper explores the foundations of this field as well as the branches of the more recent approaches including multi-lingual solutions, phrase recognition and substitution, SQL keyword mapping, and fuzzy logic applications.

We have proposed method of incorporating fuzziness in Complex query based database for accommodating fuzzy queries which are approximations based on human knowledge. Fuzzy queries to relational database are proposed as one candidate model and fuzzy relational database is presented as another relational model. With the help of example of employee Complex query based database, we have successfully implemented fuzzy logic based queries using dot net environment for GUI and SQL server for databases. This establishes feasibility of the concepts proposed.

Complex query based database systems provide an improvement on communication and data processing due to its data distribution throughout different network sites. Not only is data access faster, but a single-point of failure is less likely to occur, and it provides local control of data for users. However, there is some complexity when attempting to manage and control Complex query based database systems.

A Complex query based database allows faster local queries and can reduce network traffic. With these benefits comes the issue of maintaining data integrity.

FUZZY QUERY FETCHING

To process a user input query, the prime rung is speech labeling (tagging); subsequently word tagging. The second rung is deconstructing the tagged sentence by a grammar rules. The grammar parser examines the query sentence consistent with the tag of every word and produces the grammar trees. Eventually, the SQL translator developments the grammar tree to acquire the SQL queries. The SQL fuzzy query format may be expressed like this: SELECT <attributes> FROM <tables> WHERE <fuzzy conditions>

Obviously, the fuzzy conditions clauses should be composed by common clauses and fuzzy ones. But in fact, it is could not be realized directly. Therefore, it must be transformed equivalence precise conditions linked with the fuzzy membership function values. The first step is to convert the user fuzzy query into a simplified classic SQL query. SHOW the detail of student WHERE ((age >10 AND grad point<= 2).The ideal SQL commands could be: SELECT * FROM student WHERE age >10 AND grad point<=2;

FUZZY RULES FOR QUERY PROCESSING

Firstly, the English input (in the form of a list) is parsed by the semantic grammar, then a post processor matches table and attribute names and joins up tables if the query involves more than one table. After that the post-processor can construct the resulting SQL query and output it. The question "What is employee id with sal greater than 3000" is processed by the system as given below. The English text is converted into a list, so that the system can process it. The sentence becomes :(WHAT IS EMPLOYEE ID WITH SALARY GREATER THAN 3000) Now this list is parsed by the semantic grammar. The parser attempts to find all possible parses of the sentence, using the semantic grammar. This sentence only has one possible parse, so only one semantic representation is returned :((SELECT ID FROM (EMP (WHERE (SALARY >= 3000))))) the semantics already look similar to an SQL query. The brackets in the semantics indicate how the phrases from the sentence relate to each other. For example, the (EMP (WHERE (SALARY>= 2000))). Similarly I was built some rules for SQL query as follows:-

- SELECT name from Emp where salary>=2000;

  (Emp (where (salary>=2000)))
- SELECT salary from Emp where empname=”ashish”;

  (Emp (salary (where (empname=ashish))))
- SELECT H-marks from Result where H-marks =50;

  (Result (where (H-marks =50))))
- SELECT first_name, last_name FROM student_details;

  (Student_detail(first_name,last_name))
- SELECT * FROM student_details where name=’alok’;

  (student_details(where (name=alok)))

FUZZY SET OF QUERY PROCESSING FOR FUZZIFICATION

After completion rules, the fuzzy in the fuzzy inference engine designing, membership functions are included. Fuzzy Inference Engine Designing: Most important application of fuzzy system is in uncertain issues. When any query is having imperfect information, fuzzy logic is a suitable tool for dealing with that query. So, first step of fuzzy inference engine designing is determination of input and output variables. After that membership function for all attributes will be designed.

Input Attributes: various input attributes are :

Fuzzy Query : This input field is divided into four sets:-

- **1NF**- 1 Normal Form
- **2NF**-2 Normal Form
- **3NF**-3 Normal Form

Gender:This input field is divided into two sets:-

- Male
- Female

Age : This input field is divided into 4 fuzzy sets:-

- Youth
- Middle
- Mature
- Old

Input field	Normal forms	Fuzzy Set	Range
Age	1NF	Youth	< 18
		Middle	16-42
		Mature	38-58
		Old	>50
	2NF	Youth	< 18
		Middle	16-42
		Mature	38-58
		Old	>50
	3NF	Youth	< 18
		Middle	16-42
		Mature	38-58
		Old	>50

Table 2: Fuzzy set and Membership function of Age

$$\mu_{youth}(\alpha) = \begin{cases} 1 & \alpha \le 16 \\ (18-\alpha)/7 & 16 < \alpha < 18 \end{cases}$$

$$\mu_{middle}(\alpha) = \begin{cases} 1 & 18 \le \alpha \le 38 \\ (42-\alpha)/7 & 38 \le \alpha \le 42 \\ (\alpha-16)/7 & 16 \le \alpha \le 18 \end{cases}$$

$$\mu_{mature}(\alpha) = \begin{cases} 1 & 42 \le \alpha \le 50 \\ (58-\alpha)/7 & 50 \le \alpha \le 55 \\ (\alpha-38)/7 & 38 \le \alpha \le 42 \end{cases}$$

$$\mu_{old}(\alpha) = \begin{cases} 1 & \alpha \ge 58 \\ (\alpha-50)/7 & 50 < \alpha < 58 \end{cases}$$

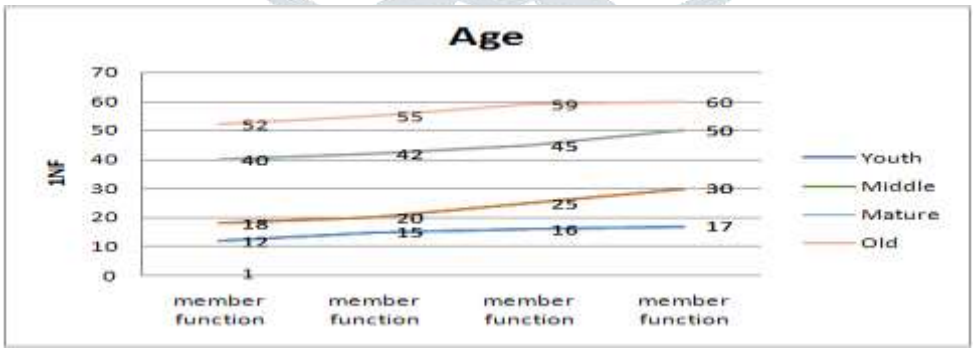

Fig. 1: Membership function of Age with Normal forms

CONCLUSIONS

Proposed work for complex query processing handles two important issues in data distribution i.e. minimizing query response time through partitions and handling fuzziness in database by translating fuzzy queries into SQL. Using the same model and same fuzzy querying algorithm, most of the general data types can be implemented by enumerating the members of domain. For more complex data types, graphics, documents etc, same formalization can be implemented. Fuzzy logic based query optimization in Complex query based database have an important impact on the performance of

Complex query based database query processing.

REFERENCES

1. G.J. Klir and B. Yuan "Fuzzy Sets and Fuzzy Logic: Theory and Applications", Prentice Hall, Inc. Englewood Cliffs, N. J., U.S.A [2001].
2. Cao, J., Kubota, N., Liu, H., "A Two Stage Pattern Matching Method for Speaking Recognition of Partner Robots", Procceedings of the 2010 Fuzz-IEEE, WCCI2010, Barcelona, Spain, 499-504, 2010.
3. Huangi,Guiang Zangi, Phillip C-Y Sheu "A Natural Language database Interface based on probabilistic context free grammar", IEEE International workshop on Semantic Computing and Systems 2008.
4. El-Fiqui, H., Petraki, E., Hussein, A., "A Computational Linguistic Approach for the Identification of Translator Stylometry using Arabic-English Text", Procceedings of the 2011 IEEE International Conference on Fuzzy Systems", Taipei, Taiwan, 2029-2045, 2011.
5. Fu, G., Wang, X., "Chinese Sentence-Level Sentiment Classification Based on Fuzzy Sets", Proceedings of the COLING2010, Beijing, China, 312-319, 2010.
6. El-Beltagy, S.R., Rafea, A., "An accuracy-enhanced light stemmer for arabic text", ACM Transactions on Speech and Language Processing (TSLP), Vol. 4, (2), Article 2, ACM, 201.1
7. Jasmeen Kaur, Bhawna chauhan &,Jatinder Kaur Korepal "Implimentation of query processor using automata and natural processing", International Journal of Scientific and Research Publications, Volume 3, Issue 5, May.