GRAPHICAL USER INTERFACE AND RELATIONAL DATABASE ACCESSING USING NLP

[1]ABHIJEET R. SONTAKKE, [2]AMIT PIMPALKAR

[1, 2]Department of Computer Science and Technology
[1, 2]G.H. Raisoni Academy of Engineering and Technology, Nagpur, India.
E-mail: [1]abhi.unicorn.g@gmail.com, [2]amit.pimpalkar@raisoni.net

Abstract- Data has been stored in the database and the databases are the major sources of information. Information is playing an important role in day-to-day life. This database technology has the major impact on the growing use of computer and internet. Database management system has been used for accessing, storing and retrieving data. However, database system is not understandable to each and every user because they are hard to use and understand. People with no knowledge of database language may find it difficult to access database. Therefore, there is need to find out the new technique and methods to access the database with the use of Natural Language Processing. Therefore this idea of using natural language instead of SQL triggered the development of a different type of processing method called Natural Language Interface to Database (NLIDB). Where user does not have any need to learn the formal language, they can give query in their native language. For the people who are comfortable with the Hindi language need this application to accept Hindi sentence as a query, process it and after execution provide result to the user in the same language which is nothing but the Hindi Language Interface to Database Management System.

Keywords- DBMS, HLIDBMS, NLP, NLIDB, SQL.

I. INTRODUCTION

The requirement of information and data is very important part of life. There are various sources of information but the major one is the databases. Database helps us to store, access and retrieve information. Any organization or industry is possible without the use of database. Each n every computer applications are dependable on database to access the information. For that it is necessary to have knowledge of formal query language like SQL but it is very difficult for everyone to learn and write SQL queries. To overcome this problem many researcher have brought out to use Natural Language (NL) i.e. English, Hindi, Marathi, Bengali, Arabic etc. in place of formal query language which can be perfect interface between an application of computer and non technical user.

1.1Natural Language Processing (NLP)

Natural language processing (NLP) is a field of computer science, artificial intelligence, and computational linguistics concerned with the interactions between computers and human (natural) languages. As such, NLP is related to the area of human–computer interaction. Many challenges in NLP involve natural language understanding, that is enabling computers to derive meaning from human or natural language input, and others involve natural language generation. In theory, NLP is a very attractive method of human computer interaction. Natural Language Understanding is sometimes referred to as an AI-complete problem because it seems to require an extensive knowledge about the outside world and the ability to manipulate it. NLP has significantly overlapped with the field of computational linguistics, and is often considered a sub-field of artificial intelligence.

The foundation of NLP lies in a number of disciplines' like computer and information sciences, linguistics, mathematics, electrical and electrical engineering, artificial intelligence and robotics, psychological, agriculture, weather forecasting etc. [1]. Applications of NLP include a number of fields of studies, such as machine translation, natural language interface to database, natural language text processing and summarization, user interfaces, multilingual and cross language information retrieval (CLIR), speech recognition, AI and expert system, and so on. [2].

1.2 Natural Language Interface to Database (NLIDB)

A person with no knowledge of database language may find it difficult to access database easily. Therefore, SQL tutor was developed for analyzing the ability of Natural Language Processing to develop products for people to interact with database in simple English. These, products have created a revolution in extracting information from databases. They have discarded the fuss of learning SQL and time is also saved in learning in query language.

II. RELATED WORK

Work for developing NLIDB has started in early seventies. Since then many systems have been developed. Early systems have many flaws then some systems were developed to overcome these flaws. Some of the developed NLIDB systems are discussed below.

Following are some developed NLIDB systems are given-

2.1 LUNAR System

W. Woods etal [5] has given information about LUNAR system that answers questions about samples

of rocks brought back from the moon. The meaning of system's name is that is in relation to the moon. To accomplish its function the LUNAR system uses two databases; one for the chemical analysis and the other for literature references. The LUNAR system uses an Augmented Transition Network (ATN) parser and Woods" Procedural Semantics. W. Woods [6] have also given the study of the LUNAR system performance which was quite impressive; it managed to handle 78% of requests without any errors and this ratio rose to 90% when dictionary errors were corrected. But these figures may be misleading because the system was not subject to intensive use due to the limitation of its linguistic capabilities.

2.2 LADDER

It was designed as a NLIDB of information about US Navy ships. According to G. Hendrix etal [7], the LADDER system uses semantic grammar to parse questions to query a distributed database. Although semantic grammars helped to implement systems with impressive characteristics, the resulting systems proved difficult to port to different application domains. Indeed, a different grammar had to be developed whenever LADDER was configured for a new application [5]. The system uses semantic grammars technique that interleaves syntactic and semantic processing. The question answering is done via parsing the input and mapping the parse tree to a database query. The system LADDER was implemented in LISP. At the time of creation of the LADDER system, it was able to process a database that is equivalent to a relational database with 14 tables and 100 attributes.

2.3 RENDEZVOUS System

In the system developed and studied by E. Codd [8] users could access databases via relatively unrestricted natural language. In this Code's system, special emphasis is placed on query paraphrasing and in engaging users in clarification dialogs when there is difficulty in parsing user input.

2.4 PLANES

D. Waltz stated [9] the Programmed Language-based Enquiry System (PLANES) at the University of Illinois Coordinated Science Laboratory. PLANES include an English language front end with the ability to understand and explicitly answer user requests. It carries out clarifying dialogues with the user as well as answer vague or poorly defined questions. This work is being carried out using database based upon information of the U.S. Navy 3-M Maintenance and Material Management, it is a database of aircraft maintenance and flight data, although the ideas can be directly applied to other non-hierarchic record-based databases.

2.5 PHILIQA

This was known as Philips Question Answering System (PHILIQA) explained by R. Scha [10], uses a syntactic parser which runs as a separate pass from the semantic understanding passes. This system is mainly involved with problems of semantics and has three separate layers of semantic understanding. The layers are called "English Formal Language", "World Model Language", and "Data Base Language" and appear to correspond roughly to the "external", "conceptual", and "internal" views of data.

2.6 CHAT-80

. CHAT-80 was implemented entirely in Prolog and it is the best NLIDBs system. It transformed English questions into Prolog expressions, which were evaluated against the Prolog database. The code of CHAT-80 was circulated widely and formed the basis of several other experimental NLIDBs. The database of CHAT-80 consists of facts (i.e. oceans, major seas, major rivers and major cities) about 150 of the countries world and a small set of English language vocabulary that are enough for querying the database [24] .

2.7 TEAM

B. J. Gross has given a paper on TEAM (Transportable Natural Language Interface system). A large part of the research of that time was devoted to portability issues. TEAM was designed to be easily configurable by database administrators with no knowledge of NLIDBs [11, 12].

2.8 ASK

Allowed end-users to teach the system new words and concepts at any point during the interaction. ASK was actually a complete information management system, providing its own built-in database and the ability to interact with multiple external databases, electronic mail programs and other computer applications. All the applications connected to ASK were accessible to the end-user through natural language requests. The user stated his/her requests in English and Ask transparently generated suitable requests to the appropriate underlying systems.

2.9 JANUS

P. Resnik studied [13] that it had similar abilities to interface to multiple underlying systems (databases, expert systems, graphics devices, etc). All the underlying systems could participate in the evaluation of a natural language request, without the user ever becoming aware of the heterogeneity of the overall system. JANUS is also one of the few systems to support temporal questions.

2.10 EUFID

M.Templeton etal [14] has given that the EUFID system consists of three major modules, not counting the DBMS. First is analyzer module, second is mapped module and third is translator module.

2.11 DATALOG

It is an English database query system based on Cascaded ATN grammar. By providing separate

representation schemes for linguistic knowledge, general 14 world knowledge, and application domain knowledge, DATALOG achieves a high degree of portability and extendibility [15]. Systems that also appeared in mid-eighties were LDC [16], TQA [17], TELI [18] and many others.

2.12 SQL Tutor

SQL can be very difficult for beginner users to understand. The SQL-Tutor program tutors students by assisting the students through a number of database questions from four different databases. A student model is kept for each student based on query constraints (each constraint represents a part of the query that is necessary to answer the question). Each time a particular query constraint is used, SQL-Tutor records whether it was used successfully or unsuccessfully. In this way a model of a student's strengths and weaknesses is generated and SQL-Tutor can select questions which re-enforce problem areas or introduce new query concepts [22].

2.13 Other Systems

B. Sujata etal [19] introduced a concept that, the SQL norms are been pursued in almost all languages for relational database management systems. However, not everybody is able to write SQL queries as they may not be aware of the structure of the database. So this has led to the development of Natural Language interface for databases. There is an overwhelming need for non-sophisticated users to query relational databases in their natural language instead of working with the syntax of SQL. As a result many NLIDB have been developed, which provides different options for manipulating queries. The idea of using Natural Language instead of SQL has prompted the development of new type of processing called NLIDB.

M. Dua etal [20] had implemented the system based on NLP which gives output on the basis of NLIDB and HLIDB management system that give the proper result for only select, update and delete queries. A. kumar [21] has given the system which is based on HLIDB using semantic matching.

III. PROPOSED SYSTEM

3.1. Problem Statement

Hindi language interface to relational database is completely based on the rules through which we are going to perform the operations like select, insert, update, delete. We are also working to provide the advance query operation such as functionality of aggregate functions such as MIN (), MAX (), SUM () and AVG (). The user will type the query in Hindi language and that natural language has been processed and will give the output in Hindi language only. Time difference has been calculated, system will give translation time and execution time in milliseconds as well as in nanoseconds.

3.2. Methodology

To achieve the above objective methodology used is given as- we are going to use the rule based system which will follow and execute each and every query as per the rules made for it. First it will identify the nature of the query i.e. select, update, delete, create, insert and also it will identify that the query is with aggregation functions or not. We are using the relational database so it is very much flexible we can easily store all Hindi as well as English values in it and also we can easily retrieve it. Randomize automatic record generation technique is also there so that we can easily generate maximum number of records in very less time. Appropriate mapping of tokens with database values should be done by extracting table, columns information from input Hindi sentences. With the help of stored values of databases generate SQL query by mapping input query. Finally we will execute the Hindi query and also get the output in Hindi language itself.

3.3 Implementation & Architecture of the system

Architecture of Hindi language interface to relational database using NLP is given and explained below from fig1.

This architecture is known as HLIDBMS i.e. Hindi Language Interface to Database management System. There are important phases i.e. Tokenizer, query type rule, query table rule, basic queries and its sub rules, query generator engine DBMS & database server.

In tokenize phase Hindi sentence is split into tokens. This is done with fact that all the tokens are separated by a space gap from each other. All the tokens which we get in this phase are stored in an array. Tokens are words of Hindi language. Token may be a table name, column name, condition, any value, command name, operation name or any non-useful word. To understand this; let the user query is as:

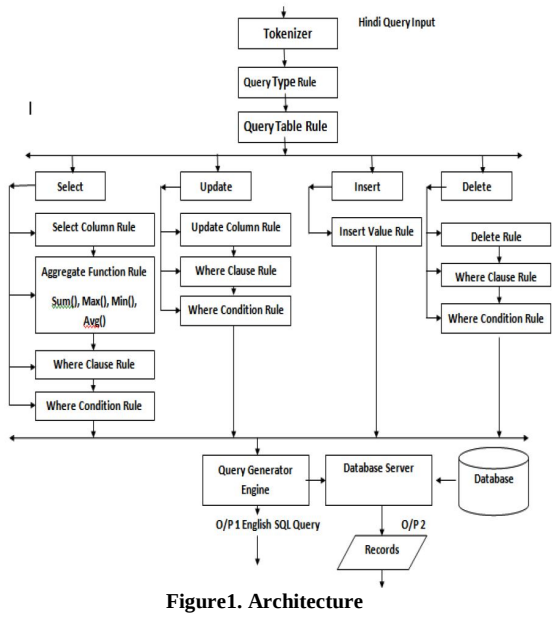

Figure1. Architecture

सभी विद्यार्थी का नाम,अंक बताओ. This Hindi sentence has 7 tokens. First token is सभी which is the starting of sentence. Now सभी means it is reflecting like select all i.e. in SQL we say 'Select *', another token is विद्यार्थी it is reflecting the name of the database table i.e. 'student table' Some tokens may be fields name as in the above query नाम and अंक are the field names. There is conjunctions also like का as well as we also included the comas (,) in the list of tokens & finally last thing is बताओ which is reflecting as the 'select query' Therefore after this step we have all the tokens from which the sentence is composed of.

After that we will apply the query type rule. Query type rule is a rule which will identify which type of query it is whether it is select, insert, update, delete type of query. We are given with the query properties through which we can easily identify the associated Hindi word which is given in the sentence within a query and is given below in figure 2.
Later it will identify the table name with the help of query table rules. It will just see whether the given table is present there or not. These both the things have been possible because of the tokenizer and its tokens which we are matching under each rule. Once the query rules and table rules has been applied then we will proceed with the further tokens and we will apply the sub rules of the selected query.
If the query in Hindi will be the select query then it will look for the rules like column rules, aggregate function rule, where clause and where condition rule.

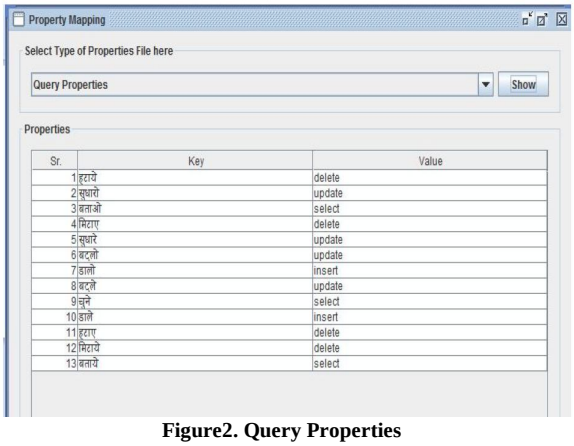

Figure2. Query Properties

It will work with the help of tokens only. It is like column rules it will select the number of columns given in the Hindi query.
Aggregate function will identify whether it is min (), max (), sum (), avg () query or not . Other rules like where clause for that we are given with the properties i.e. It will identify all the associated Hindi English words which are given below in the fig 3.

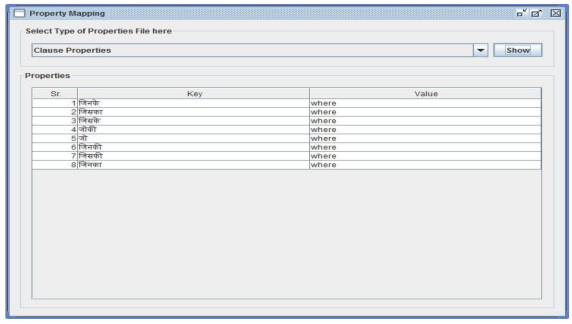

Figure3. Where clause property

Similarly where condition is also there it will work like same as given above it is consisting of all the conditional part and its associated Hindi words including <,>,=,logical and ,or not etc.
Similarly for update query it is having update column rule ,where clause rule and where condition rule and its working is same as explained above. The same way insert and delete also work.

At last there is query generator which will generate query from Hindi sentence .that query generated will be fired to database and all the selected records selected rows has been displayed in Hindi Language.
SQL is generated in this phase according to Hindi sentence. Execute query and display result to user the above SQL query is executed and result of which in Hindi language is displayed to user. The output is in the form of Hindi language and we are giving query also in Hindi language and processing of all this has been done by inner module as explained above.

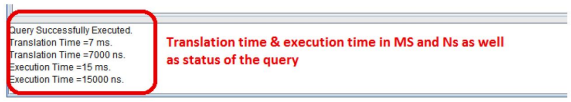

Figure4. GUI & timing results

Once the query has been executed and the result has been shown side by side it will also show the timing result which include whether the query has been successfully executed or not if it is failed it will show the unsuccessful message as shown in the fig4. It will also give the translation time in milliseconds as well as nanoseconds to notice the minute difference during conversion and same in the case of execution time also, it will show the time required to execute the query.

CONCLUSIONS

Rule based graphical user interface to relational database is presented in this paper. The system will accepts Hindi sentence as a query and gives output in Hindi itself. It is very much useful for the people who do not have any prior knowledge of database and SQL queries languages. We are using different rule along with the NLP to perform operation such as insert, update, delete, select as well as the aggregate

functions such as min (), max (), sum (), avg () etc. This system can be enhanced by making it more generic. We can also implement it for very complex queries like join operations & order by operations (queries). To make the system more friendly the dialogue based system can be used in which user will provide the input Hindi query through speech interface.

REFERENCES

[1] Akshar Bharthi, Y. Krishma Bhargava and Rajeev Sangal, "References and Ellipsis in an Indian Languages Interface to Databases", Journal of Computer Science of India, VOL 23, no. 3, pp 60-82, Sep. 1993.

[2] Gobinda G. Chowdhury, "Natural Language Processing ", Annual Review of Information and Sciences Technology, VOL 37, no. 1, pp 51-89 ,2003.

[3] Amandeep Kaur, "Punjabi Language Interface to Database", ME Thesis, Thapar University, Jun 2010.

[4] H. R. Tennant, K. M. Ross, M. Saenz, C.W. Thompson and J.R. Miller, "Menu Based Natural Language Understanding", Proceeding of the 21st Annual Meeting of ACL, Cambridge, Massachusetts, pp. 151-158, 1993.

[5] W. Woods, R. Kaplan and B. Webber, B., "The Lunar Sciences Natural Language Information System", Final Report. B. B. N. Report No 2378, USA, 1972.

[6] W. Woods, "An experimental parsing system for transition network grammars. In Natural Language Processing", Algorithmic Press, New York, USA, 1973.

[7] G. Hendrix, E. Sacrdoti, D. Sagalowicz, and J. Slocum, "Developing a natural language interface to complex data", ACM Transactions on Database Systems, Volume 3, No. 2, pp. 105 – 147, USA, 1978.

[8] E.F. Codd, "Seven steps to rendezvous with the casual user", In IFIP Working Conference Data Base Management, pp 179–200, 1974.

[9] D.L. Waltz., "An English Language Question Answering System for a Large Relational Database", Communications of the ACM, pp 526– 539, 21(7): July 1978.

[10] R.J.H. Scha., "Philips Question Answering System PHILIQA1", In SIGART Newsletter, no.61. ACM, New York, February 1977.

[11] B.J. Grosz, "TEAM: A Transportable Natural-Language Interface System", In Proceedings of the 1st Conference on Applied Natural Language Processing, Santa Monica, California, pp 39–45, 1983.

[12] B.J. Grosz, D.E. Appelt, P.A. Martin, and F.C.N. Pereira, "TEAM: An Experiment in the Design of Transportable Natural-Language Interfaces", Artificial Intelligence, pp 173–243, 32: (1987).

[13] P. Resnik, "Access to Multiple Underlying Systems in JANUS", BBN report 7142, Bolt Beranek and Newman Inc., Cambridge, Massachusetts, September 1989.

[14] M. Templeton and J. Burger, "Problems in Natural Language Interface to DBMS with Examples from EUFID", In Proceedings of the 1st Conference on Applied Natural Language Processing, Santa Monica, California, pp 3–16, 1983.

[15] C.D. Hafner, "Interaction of Knowledge Sources in a Portable Natural Language Interface", In Proceedings of the 22nd Annual Meeting of ACL, Stanford, California, pp 57–60, 1984.

[16] B.W. Ballard, J.C. Lusth, and N.L. Tinkham, "LDC-1: A Transportable, Knowledge based Natural Language Processor for Office Environments", ACM Transactions on Office Information Systems, pp 1–25, January 1984.

[17] F. Damerau, "Operating statistics for the transformational question answering system", American Journal of Computational Linguistics, pp 30–42, 7: 1981.

[18] B. Ballard and D. Stumberger, "Semantic Acquisition in TELI", In Proceedings of the 24th Annual Meeting of ACL, New York, pp 20–29, 1986.

[19] B. Sujata, S. Viswanadha Raju and Humera Shaziya, "A Servey of Natural Language Interface to Database Management System" International Journal of Science and Advance Technology", vol.2, no. 6,june 2012 .

[20] Mohit dua, Sandeep Kumar, Zorawar Singh Virak, "Hindi Language Graphical User interface to Database Management System" 12th International Conference of Machine Learning and Application 2013.

[21] Mohit dua, Sandeep Kumar, Zorawar Singh Virak, "Hindi Language Graphical User interface to Database Management System" 12th International Conference of Machine Learning and Application 2013.

[22] Seymour Knowles, "A Natural Language Database Interface For SQL-Tutor", 1999.

[23] Abhijeet R. Sontakke, Prof. Amit Pimpalkar "A Review Paper on Hindi language Graphical User Interface to Relational Database using NLP" International journal of Advanced Research in Computer Engineering & technology (IJARCET)volume 3 Issue 10, October 2014 .

[24] M. E. Saleh, "Semantic Based Query in Relational Database Using Ontology", Canadian Journal on Data, Information and Knowledge Engineering, vol.2, 2011.

★★★