An Advanced Natural Language Interface to Databases

Jaya Sharma
PG Scholar, CSE
BMSCE Muktsar

Khushdeep Kaur
Assistant Professor,CSE
BMSCE Muktsar

Abstract: Database management systems have been widely used for storing and retrieving data. However, databases are difficult to use and there interface is complex and the same is difficult to access. To make it easy for a user to retrieve data, an interface is developed in which a database can be accessed by a user through querying in Hindi language and to get the result in same language. In order to develop an improved Hindi language graphical user interface to database management system. The proposed system can handle single and multiple columns retrieval queries, selection of whole table, conditional queries (between, in), join queries and queries that include nested, functions and logical operators. Since a user should not be able to update or delete data from database so the user is suggest on selection queries.

Keywords: *Natural Language Processing System, SQL, Hindi Language interface to Relational Database, Natural Language Interface to Relational Database.*

*****

I. INTRODUCTION

We require information in our daily life. One of the major sources of information is database. Database management systems have been widely used for storing and retrieving data. However, databases are difficult to use and there interface is complex and the same is difficult to access. To make it easy for a user to retrieve data, an interface is developed in which a database can be accessed by a user through querying in Hindi language and to get the result in same language. Since existing system had some limitations so we removed those drawbacks of existing system by adding valuable features. In order to develop an improved Hindi language graphical user interface to database management system a STUDENT database is identified as a case study. The database has two tables Student and Department which contains the information of students and departments respectively, both tables have a common attributes. A query can be done in two ways either by typing or selecting the query from stored tested queries. Tested queries are those queries that have been executed earlier and stored by user after their execution. The proposed system can handle single and multiple columns retrieval queries, selection of whole table, conditional queries, join queries and queries that include functions and logical operators. Since a user should not be able to update or delete data from database so the user is suggest on selection queries.

Using Java Swing, a graphical user interface has been designed for the system. SQL Server has been used as database and it is connected to java using Java database connectivity. All the data values stored in database are in Hindi language. This is done by using Unicode character set. Hindi Shallow parser is used to parse the input Hindi sentence. It is used to tokenize the Hindi input query. The tested queries contain almost all type of selection queries.

II. LITERATURE REVIEW

There are many systems, the best known NLIDB of sixities and early seventies were developed which can be summarized as below:

ISSN: 2321-8169
3200 – 3203

Table 1. Summary of Existing System

S.NO	System Name	Domain	Language	Approaches	Techniques	Year
1	LUNAR [2]	samples of rocks Brought back from moon	English-sql-English	Connectionist (neural network)	Semantic Grammar System	1973
2	RENDEZVOUS	General	English-sql-English	Dialogue based	–	1977
3	PHILIQA [6]	General	English-sql-English	–	Syntactic Grammar System	1977
4	LADDER [3]	US-Navy ships	English-sql-English	Corpus Based	Semantic Grammar System	1978
5	CHAT-80 [4]	General	English-Sql-English	Dialogue Based	Semantic Grammar System	1980
6	TEAM [7] [8]	General	English-Sql-English	–	Semantic Grammar System	1987
7	JANUS [9]	General	Natural languages	Menu based	ER-based intermediate representation	1989
8	ASK	Complete information management system	English-Sql-English	Dialogue based	–	1996
9	Intelligent tutoring system	SQL tutor Guide	English-sql-English	Dialogue based	–	1998
10	GINLIDB [13]	Natural	Natural	Lexical Analysis	Syntactic Grammar	2009
11	PNLIDB	Agriculture	Punjabi-Sql-punjabi	Shallow parser	Pattern matching	2010
12	HNLIDB	Employee	Hindi-sql-Hindi	Shallow parser	Pattern matching	2011

III. ARCHITECTURE OF NLIDB SYSTEMS

The architecture of interface to database using Hindi language is composed of four phases. The phases are given below-

- To tokenize the input Hindi sentence.
- Map the tokens with lexicon which store all the tokens and their corresponding English.
- Formulate SQL query with the help of query generator.
- Execute the query and display result on interface to user.

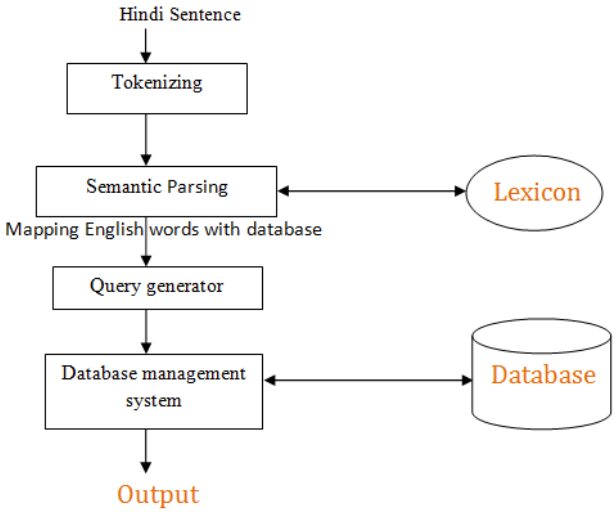

Fig. 1 Model of Existing System

Detail of each phase is described below.

Tokenize input Hindi sentence: In this 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.

Semantic Parsing *:* Lexicon store all the Hindi tokens, their corresponding English word and type of token whether it is table name, column name, any value, operation, command or something else. Tokens which we extracted in above step are matched with the tokens stored in lexicon one by one. If it matches then its corresponding English word is saved along with its type. This is the most important phase. All the useless tokens discarded in this phase only useful tokens are stored. After this step we will have with table name, field name (columns name), conditions, function etc. which will be further used to make SQL query.

Formulate SQL query: Now in the beginning of this phase we are with table name, column name, condition and command etc. SQL is generated in this phase according to Hindi sentence.

Execute query and display result to user: In this phase the above get SQL query is executed and result of which in Hindi language is displayed to user on the interface.

- **Tokenizing :** Tokens are gathered from the input sentence by using the logic that all the tokens are separated by a gap from their adjacent tokens. The above sentence has 15 tokens. Some of them are useless that have no further use. So on end of this step we will have all the tokens. These 8 tokens are -

उन सभी विधार्थीयो के नाम बताओ जिनके अंक 50 से 60 के बीच मे है ।

- **Discard the useless tokens :** To find out the nature of tokens whether it is table name, field name, condition, command, operation or any useless token we create a database TOKEN which has all the tokens that can be given by user for our STUDENT database. This database has four fields id, Hindi_token, English_token, type. Type tells about the nature of token.
- **Map tokens to table name:** In this step we store all the tables name in English.

In our example query 'विधार्थीयो' " is the table_name.

Table_name.english = STUDENT. So

the STUDENT is saved as table name.

- **Map tokens for Function, Fields:** Field string may have single column or may have multiple columns and these columns may belong to same table or may reside in different tables of database therefore we would have need of joining operation if fields are in different tables. In our example Field String are नाम

Output of this phase will be

table_name.column_name.english.

Student.name

- **Map the token with command name:** We store this token in command string. Command can be insert, select, update or delete. Command name in our input example query is „ बताओ ". We store the equivalent English. Command.english = Select
- **Map token for condition and operation:** For our example query अंक 50 से 60 के बीच" are stored in temp_list. In our example sentence these array list have following values.

Column_arraylist- marks, Condition_arraylist- BETWEEN, logicalop_arraylist- and Value_arraaylist-50,60

Therefore output of this phase is **marks BETWEEN 50 and 60**

- **Create SQL query:** we create the SQL query in this way

"command" "column(s)name" from "table_name" where "condition"

Therefore for our example sentence SQL query is-

Select student.name from student where marks BETWEEN 50 and 60

- **Execute SQL query:** The SQL query is excuted on the database.The output is provided to user on user interface in hindi language.
- **Error messages :** If the query given by user doesn't contain sufficient tokens to be converted into a SQL query then a message is provided to the user says syntax error in query please rephrase the query. This will help the user to understand that it is linguistic failure.

IV. CONCLUSION

This system accepts query in Hindi Language that is translated into SQL query by mapping the Hind language words. Our proposed work removes the problem of existing system of not supporting functions, joining, logical operators, queries that include nested, execute queries that include group by, order by,Condition

(Between, in) successfully.

References

[1] 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, 1985.

[2] W.A Woods , R.M. Kaplan , and B.N. Webber, "The Lunar Sciences Natural Language Information System: Final Report", BBN Report 2378, Bolt Beranek and Newman Inc., Cambridge, Massachusetts, 1972.

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

[4] D. Warren and F.Pereira, " An eficent and easily adaptable system for interpreting natural language queries in Computational Linguistics" Vol.8,pp.3-4,1982.

[5] Ana-Maria Popescu, Alex Armanasu, Oren Etzioni, David Ko and Alexander Yates, "Modern Natural Language Interfaces to Databases: Composing Statistical Parsing with Semantic Tractability", COLING, 2004.

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

[7] 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.

[8] 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, Vol.32,pp.173-243,1987.

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

[10] Mrs. Neelu Nihalani, Dr. Sanjay Silakari and Dr. Mahesh Motwani "Natural Language Interface for Database: A Brief Review", International Journal of Computer Science Issues, vol. 8, Issue 2, March 2011.

[11] Himani Jain and Parteek Bhatia "Hindi Language Interface to Databases", Journal of Global Research in Computer Science, vol.2, no.2, pp. 107-112, April 2011.

[12] M. R. Joshi, R. A. Akerkar "Algorithms to Improve Performance of Natural Language Interface", International Journal of Computer Science and Applications, vol. 5, No. 2, pp. 52-68, 2010.

[13] A.Faraj EI-Mouadib, S.Zubi Zakaria, A. Ahmed Almagrous and S. Irdess EI-Feghi "Generic Interactive

[14] Interface to Databases", International Journal of Comoters issue 3, Vol.3, 2009.