International Journal of Recent Technology and Engineering (IJRTE)
ISSN: 2277-3878, Volume-3, Issue-1, March 2014

Challenges and Implementation Steps of Natural Language Interface for Information Extraction from Database

Anuradha Mohite, Varunakshi Bhojane

Abstract: ***The use of computers has started from last few years and the phase of bringing awareness in different parts of society regarding use and benefits of computer increased successfully. Not only industry but educational institutes, service sectors, manufacturing sectors etc. are using computers to store, process and update the information. Information is playing an important role in everyone's lives. One of the important sources of information is database. The huge amount of data is stored in databases. For accessing information from database by the user who is not master in SQL or database query language, there is need of Natural Language Interface. Some challenges in implementing Natural Language Interface to Database along with implementation steps have been discussed in this paper.***

Index Terms: Ambiguities, MySQL, Natural language interface to database, Structured Query Language (SQL).

I. INTRODUCTION

Natural languages are languages which human uses for communication. Computer cannot understand human languages as it is. So there is need to process natural language input. Natural language processing is a field which deals with processing user input to make it understood by computer. The main aim of Natural Language Query Processing is to interpret English sentence correctly with respective action taken.

Retrieving information from database requires knowledge of database languages like SQL. However, everybody cannot write SQL queries. So the concept of developing Natural Language Interface to Database is evolved.

Natural language interface to database is the system where user can type his / her query in natural language (English) and system will perform operation on the query to retrieve result from database.

II. CHALLENGES AND ISSUES IN IMPLEMENTING SYSTEM

Some challenges in implementing natural language interface are described below:

1. Ambiguities

Many different interpretations exist for single word or expression from a natural language input and not always one can get sufficient information for query processing.

Revised Manuscript Received on 30 March 2014.
* Correspondence Author
Anuradha Mohite*, M.E. Student, Computer Engineering Department, PIIT, New Panvel, Mumbai University, New Panvel, India.
Prof. Varunakshi Bhojane, HOD and Asst. Prof. of Computer Engineering Department, PIIT, New Panvel, Mumbai University, New Panvel, India.

Hence, the basic problem in implementing natural language interface to database is **ambiguities**, and there is need of metadata of respective database and specific question to solve this problem.

E.g. which is smallest state in India?

The above question is ambiguous as 'smallest' adjective does not specify whether it is related to area of the state or the population of state or smallest based on any other category.

E.g. how many employees are there from New York?

In this kind of question, it is not clear whether New York refers to city or the State.

E.g. where is Jammu and Kashmir?

In this question it is not clear that whether Jammu and Kashmir is single place or two different places

Incomplete input could still cause ambiguities if the systems do not possess sufficient metadata to make sense from the input by user. It is not possible for any natural language interface to produce accurate output for all kind of questions posed by users. Even human sometime misunderstand other humans.

2. Metaphor and metonymy in question

Metonymy [8] is a figure of speech where a thing is not called by its own name but by the name of something associated in meaning with that thing.

Metonymy example [8], "The book is moving right along," In this sentence *book* refers to the process of writing or publishing. Metaphor example as "fishing for information" means person is searching information not actually 'fishing'.

3. Spelling mistakes in question

Spelling errors e.g. writing 'there' instead of 'their', 'too' instead of 'to', 'who' instead of 'whom', 'begining' instead of 'beginning' etc. There is need to implement a spell check algorithm to solve this problem.

4. Handling vagueness in sentence

Vagueness in sentence may differ depending upon the nature of user using the system. E.g. suppose there is a University database interface, so in this case user can be a student or faculty member or librarian or principal. So nature of question varies according to the user.

5. Testing over all possible form of natural language

System must be tested over all possible forms of natural language for getting accurate output. Consider natural language queries as follows:

a. Who are our employees?
b. Tell me all information about our employees.
c. Give all employee details.

d. Show employee names along with other details.
All above examples are different forms of natural language input but resulting query is the same as 'SELECT * FROM EMPLOYEE' considering EMPLOYEE is table name. So system should give same result for all possible forms of input.

6. Mapping the meaning of query

Mapping the meaning of natural language input to SQL query should be taken care of.

7. User assumes intelligence

User of the system assumes system is intelligent and understands any type of question.

8. Insufficient information in query

Sometimes natural language input do not contain sufficient information which can be useful for further query processing. E.g. "Give location of ABC." So in above example it is not given whether ABC is person name or place name or city name etc.

III. PROPOSED SYSTEM ARCHITECTURE

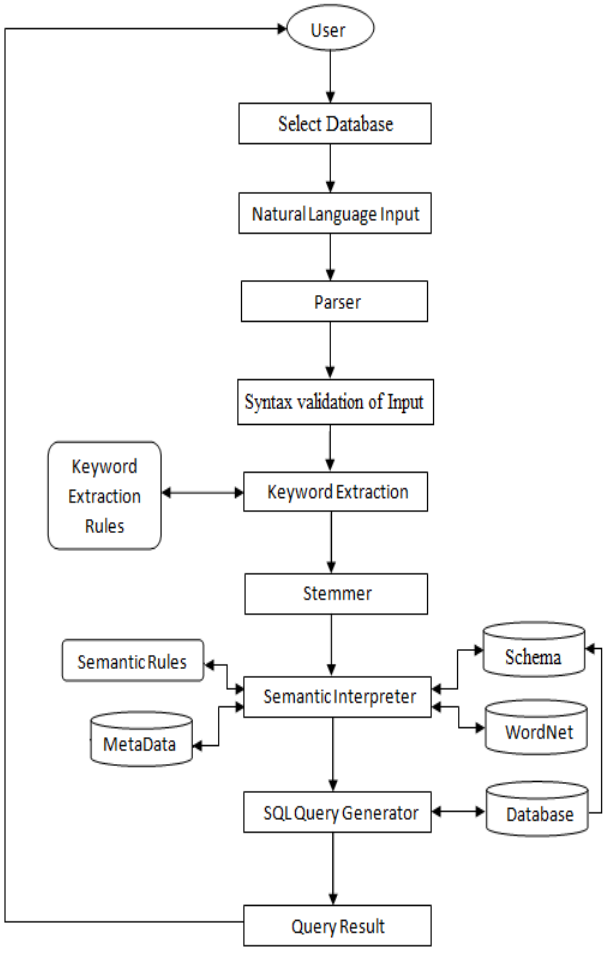

Fig.1 Architecture of Natural Language Interface System

Proposed system architecture is shown in fig.1. First step is to choose database. Database can be MySQL or MS Access having multiple tables stored in it. Question posed by user in natural language (English) is input to the system. User's query in natural language is then passed to NLP Parser. NLP Parser is a tool which can generate parse tree of given sentence. Parse tree is the hierarchical structure of user input with respective tags associated with every word in sentence. There are many parsers available as openNLP Parser, Stanford Parser, GATE etc. Consider an example as, *'what is salary and address of employee whose name is Anjali.'*
Parse tree of above query obtained using parser is as follows:
(TOP (SBARQ (WHNP (WP what)) (SQ (VP (VBZ is) (NP (NP (NN salary) (CC and) (NN address)) (PP (IN of) (NP (NP (NN employee)) (SBAR (WHNP (WP$ whose) (NN name)) (S (VP (VBZ is)))))))) (. Anjali)))
Example.2 'Display salary details of all employees'
Parse tree: (TOP (NP (NP (NNP Display) (NN salary) (NNS details)) (PP (IN of) (NP (DT all) (NNS employees)))))
The output from the parser is passed to further step where syntax validation of input is carried out. Keyword extraction is performed using pre-defined keyword extraction rules. Important tokens from sentence are the nouns, adjectives, proper nouns, verbs. In keyword extraction tokens with NN/NNS, JJ/JJR/JJS and NNP are extracted and stored in different arrays. Obtained keywords then passed to stemmer to get stem value of each keyword.
The next step is to use WordNet [2] to get synonyms of each keyword. The WordNet is used to choose correct attribute value from database. E.g. suppose there is one attribute as 'SALARY' in 'EMPLOYEE' table, but user has entered 'remuneration' or 'wage' in question then system uses synonyms set to find the matched column name from table. Then semantic interpreter uses Metadata for semantic matching. Token translation rules are used for mapping. E.g. Maximum, max, most etc. words in input should map with 'MAX' in SQL query, Total to 'SUM', minimum to 'MIN'. Some words or group of words are mapped with symbols in SQL query. E.g. 'greater than equal to' to '>=', 'Less than' to '<' etc.
Semantic interpreter make use of predefined patterns which are developed using POS tags.
Some rules used by semantic interpreter are listed below:
1. For adjective in question
Adjective is represented by JJ/ JJR/ JJS tags.
Pattern is:
Whenever encounter JJ/JJR/JJS in question e.g. greater than, greater than equal to, less, less than etc.
Find previous noun (NN/NNS), Find next CD
Use 'NN <Symbol for JJR> CD' in WHERE clause of SQL query.

2. For proper nouns in question

Proper nouns are represented by NNP tag.
Pattern is:
Whenever encounter NNP e.g. person name, city name etc.
Find previous NN/ NNS
Use 'NN/NNS = NNP' rule in WHERE clause of SQL query.
E.g. show employees from City Pune
Pune- NNP, City- NN.
So according to above rule we will get 'City = Pune' in WHERE clause of SQL query.

3. For IN BETWEEN operator

Pattern is:
Whenever encounter in between, lies in between, in range of etc. in sentence followed by values, map it with 'BETWEEN' word in SQL query.
Find next CD value e.g. 200 and 300.
Find previous noun (NN).

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)

Use 'NN BETWEEN CD and CD' in SQL query.

4. For IN operator

E.g. select all customers from city "Paris" or "London"
Find NNP Values.
Find noun related to NNP which is NN.
Use 'NN IN (NNP, NNP)' in SQL query.

Sometimes some proper nouns i.e. name of person, name of place etc. are represented by NN i.e. noun tag by parser hence in semantic interpretation step some other rules are used to identify those names and then those words are used as NNP in further rules. Some input may contain adjective, nouns, proper nouns and 'In between' operators then above rules must use in combination. After getting attributes for WHERE clause the noun which is similar to table name (in obtained schema set) goes in FROM clause and remaining nouns are used in SELECT clause of SQL query. In semantic interpretation step n-gram algorithm is used to map the words with the schema of database to get attribute names. The main condition is column names of database should be real and unambiguous. E.g. for 'SALARY'- salary, emp_salary, wage, emp_payment etc. are acceptable but sal, emp_sal, etc are not accepted by system. Outputs of semantic interpretation step are attributes which can be used to generate SQL query, which is the input to SQL query generator module. SQL query generator creates proper SQL query and apply it on database. Database used for implementing this system is MySQL and MS Access. The output from database is the final result of query which is shown to user on the screen.

IV. RECENT RESULTS

Screenshot of natural language interface tool is shown in Fig.2. There is a text field provided to enter query in English. When user clicks "Process" button further processing is carried out. Generated parse tree of given input is displayed in text area. Database used in this case is MySQL. Query is then applied on database to get final output. Query result is shown in table format. Different kinds of queries are tested on this system. Screenshots of queries with respective output is shown in fig.2 to fig.5.

There is 'Employee' table present in database having column names as name, address, city, salary, state etc. Queries entered by user shown in screenshots are as follows:

1. Display names and salaries of all employee (fig.2)
2. Show me address and salary of employee whose name is Anjali (query with proper noun) (fig.3)
3. What is maximum salary paid to an employee (Query with aggregate function) (fig.4)
4. What are names of employee having salary 20000 (fig.5)

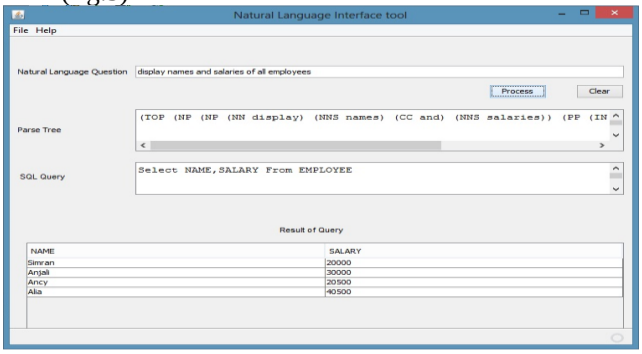

Fig.2 Screenshot of query with result from database

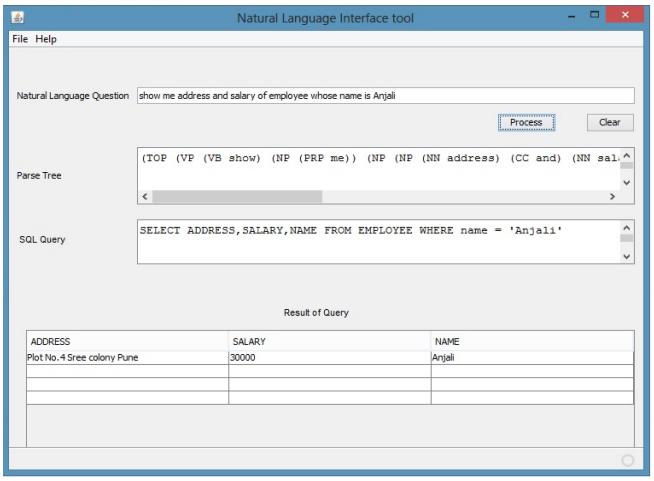

Fig.3 Condition mapping and query result

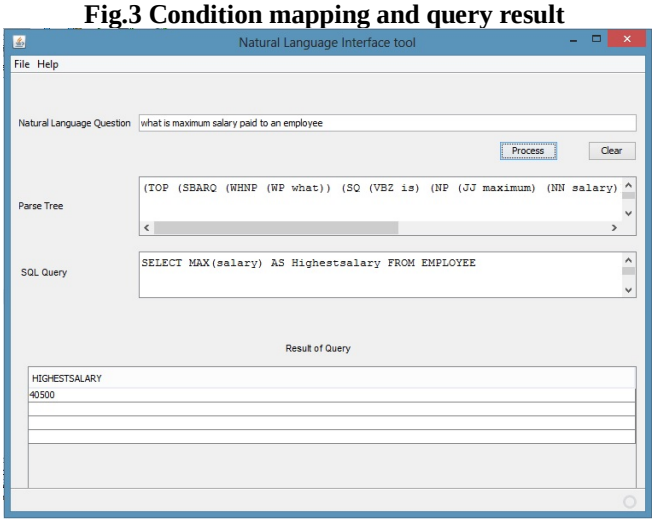

Fig.4 Query with aggregate function with result

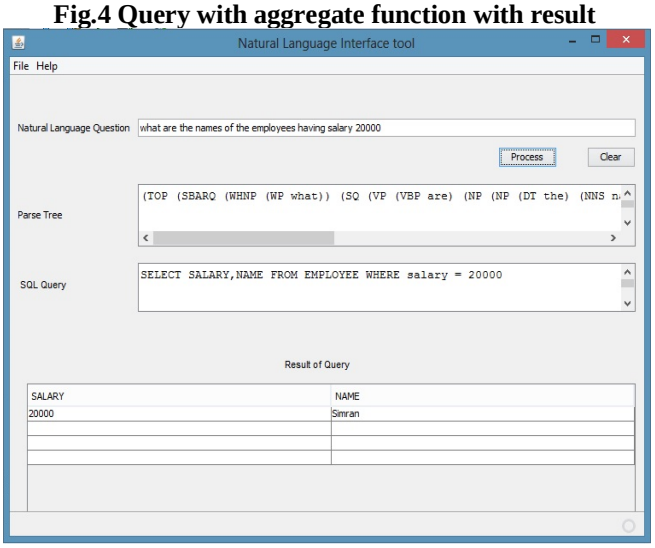

Fig.5 Condition having integer value and query result

V. APPLICATION OF THE SYSTEM

Natural language interface systems are useful in wide variety of areas as, schools and other educational institutes, government offices, hospitals, service sectors, manufacturing sectors, banks etc.

Published By:
Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP)

VI. CONCLUSION

Developing natural language interface is very important as it provides facility to access computer to all members of society. Those who are not master in database query languages can access information from database using query in natural language. This system convert natural language input to SQL query. There are many challenges in developing natural language interface to database system and some of them are explained with examples along with system implementation steps. Proposed system can be configured with other databases.

REFERENCES

[1] N Nihalani *et al:* "Natural language Interface for Database: A Brief review", IJCSI International Journal of Computer Science Issues, Vol. 8, Issue 2, March 2011

[2] G. Miller, "WordNet: a lexical database for English, Princeton University", Princeton, New Jersey, USA, 1995.

[3] Prof. Amisha Shingala, Dr. Paresh Virparia, "Enhancing the Relevance of Information Retrieval by Querying the Database in Natural form", 2013 International Conference on Intelligent Systems and Signal Processing (ISSP)

[4] Androutsopoulos, Ritchie & Thanisch P, "MASQUE/SQL- An efficient and portable Natural Language Query Interface for Relational Database", Proc of sixth International Conference on Industrial & Engineering Applications of Artificial Intelligence & Expert System, Edinburgh, 1993

[5] Gauri Rao et al., (2010) Natural Language Query Processing Semantic Grammar. (IJCSE) International Journal of Computer Science and Engineering Vol. 02, No. 02, 2010, 219-223.

[6] Karande N. D., and Patil G. A., (2009), Natural Language Database Interface for Selection of Data Using Grammar and Parsing, World Academy of Science, Engineering and Technology.

[7] F.Siasar djahantighi, M.Norouzifard, S.H.Davarpanah, M.H.Shenassa, "Using Natural Language Processing in Order to Create SQL Queries", Proceedings of the International Conference on Computer and Communication Engineering 2008 May 13-15, 2008 Kuala Lumpur, Malaysia

[8] http://en.wikipedia.org/wiki/Metonymy

Published By:
Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP)