Natural Language Interface To Database

Kulkarni Chaitali [1], Divate Prachi [2], Kale Ashwini [3], Kate Aakanksha[4]

[1]Kulkarni Chaitali Computer Engineering student, MCERC, Nashik
[2] Divate Prachi Computer Engineering student, MCERC, Nashik
[3] Kale Ashwini Computer Engineering student, MCERC, Nashik
[4]Kate Akanksha Computer Engineering student, MCERC, Nashik

---***---

Abstract - *Information is playing an important role in our lives. One of the major sources of information is databases. Databases and database technology are having major impact on the growing use of computers. Almost all IT applications are storing and retrieving information from databases. There are various interfaces available to retrieve data such as form based, natural language and keyword based. Data retrieval from the database requires knowledge of database language like SQL [1].*

In this paper we have proposed architecture of a Natural Language and Keyword Based Interface for Database (NLKBIDB) which provides solution for syntactically correct and incorrect natural language input query. Our partial experiment of Lexical Analyzer and Keyword based interface on agriculture survey database solves 53% of syntactically incorrect query which will not be solved by natural language interface resulting in increase of rate of SQL query conversion.

Key Words: Natural Language Interface for Database (NLIDB), Keyword Base Interface to Database (KBIDB).

1. INTRODUCTION

Databases are gaining prime importance in a huge variety of application areas employing private and public information systems. Databases are built with the objective of facilitating the activities of data storage, processing, and retrieval associated with data management in information systems. There are diverse solutions of user interface to retrieve the data from database i.e. Form based interface, Natural language based query interface, Keyword based query interface. Form based interface provides the interactive user interface but it gets failed if provider has not provided retrieval of data variations which is possibly manipulated by SQL. Natural language or Keyword based interface allow the user to access data by entering query in natural language either in English or in any other language. Main objective of NLIDB is to accept the query sentence and try to understand it by applying lexicon, syntactic and semantic analysis and then converts into SQL [1]. Keyword based interface accepts query type of search engine query which retrieves keyword from the input query and converts into SQL by applying rules from the generated knowledge base. Researchers have worked on Natural Language & Keyword based Interface to convert input query into SQL query independently. NLIDB provides accuracy for generating a SQL by using the SQL generation rule based on Natural language concept but it fails to generate SQL of syntactically incorrect query while KBIDB handles syntactically correct and incorrect query both but SQL generation rules of KBIDB is less strong than NLIDB. So, we have proposed the NLKBIDB architecture on combined approach of NLIDB and KBIDB for relational database which considers the strength of both natural language and keyword based interface of syntactically correct and incorrect query. It will apply the generation rules of NLI for syntactically correct query to get more accurate SQL query but for incorrect query it will apply rules of KBI for SQL conversion to improve the accuracy. NLKBIDB tokenizes the natural language query by lexical analyzer and Lexicons are parsed by the syntax analyzer. If Input query is syntactically valid, Lexicons will be analyzed by the semantic analyzer using domain ontology and then SQL Generator will be used to generate SQL. But if Syntax analyzer gets failed to validate the Query then keyword based agent will convert it into SQL by using rule base knowledge.

1.1 Intelligent database system

An IDBS is endowed with a data management system able to manage large quantities of persistent data to which various forms of reasoning can be applied to infer additional data and information. This includes knowledge representation techniques, inference techniques, and intelligent user interfaces – interfaces which extend beyond the traditional query language approach by making use of natural language facilities. [2, 3]

These techniques play important role in enhancing databases systems : knowledge representation techniques allow one to represent better in the DB the semantics of the application domains, inference techniques allow one to reason about data to extract additional data and information. The research and advancement of NLIDB, an important step towards the development of intelligent databases system and it has emerged as a new discipline and have fascinated the attention to number of researchers [1].

1.2 Natural language interface to databases

Natural Language Interfaces is a hot area of research since long. The purpose of Natural language Interface to Database System is to accept requests in English or any natural language and attempts to 'understand' them or we can say that Natural language interfaces to databases (NLIDB) are systems that translate a natural language sentence into a database query.[4]

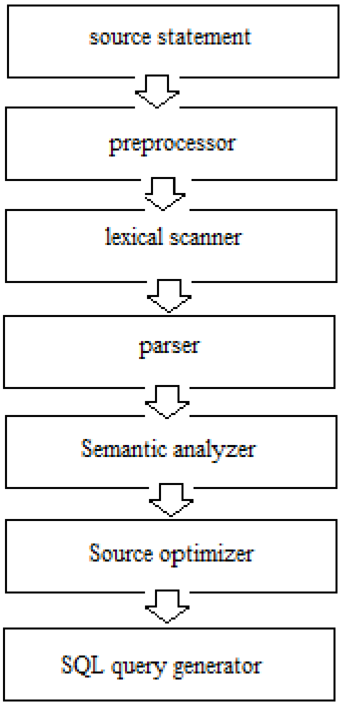

Fig: NLIDB

a. Source Statement
It is the input given by the user to the NLDBI System which includes the English language statements in the form of WH type questions.

b. Pre-processing
Pre-processing plays an important role in many NLP Applications. It includes following process
1) Tokenization/Lexical Scanner
2) Text Normalization
3) Tagging

c. Parser
It is the process of assigning the structural description to the sequence of words in NL. There will be zero description for ungrammatical sentence.

d. Semantic Analyzer
A semantic grammar system is very similar to the syntax based system, meaning that the query result is obtained by mapping the parse tree of a sentence to a database query. The basic idea of a semantic grammar system is to simplify the parse tree as much as possible, by removing unnecessary nodes or combining some nodes together.

e. Source Optimizer
Optimize the user input to generate the SQL Query for most relevant information extraction and retrieval.

f. Query Generator
After the successful parsing of the statement given by the User, the system generates a query adjacent to the user Statement in SQL and further it return to the end user of Database. [2]

1.3 Algorithm

Based on the study of patterns of questions and the sentences having semantic symmetry. We have formulated rules that pick up exact sentences as answer among from number of candidate answer sentences. [6]Following algorithm shows how the problem caused by semantic symmetry is solved with only shallow parsing applied on the corpus.

Two important factors considered are:
1. Sequence of keywords in question and in the candidate answer sentence

2. POS of keywords
The algorithm scans each candidate answer sentence and applies following rule to check whether that sentence is correct answer sentence or not.

Rule 1
If (sequence of keywords in question and candidate answer matches) then
If (POS of verb keyword are same) then
Candidate answer is Correct

Rule 2
If (sequence of keywords in question and candidate answer do not match) then

If (POS verb keyword are not same) then
Candidate answer is Correct

Otherwise
Candidate Answer is wrong

Explanation of Shallow base with Example

A "parser" is a system that transforms sentences (strings of characters) into a representation that describes the groupings of words (phrases) and their relations (e.g. subject and object). The representation of choice for such information is a syntactic tree in which nodes refer to phrases, word categories, or words, and links refer to relations between these objects:

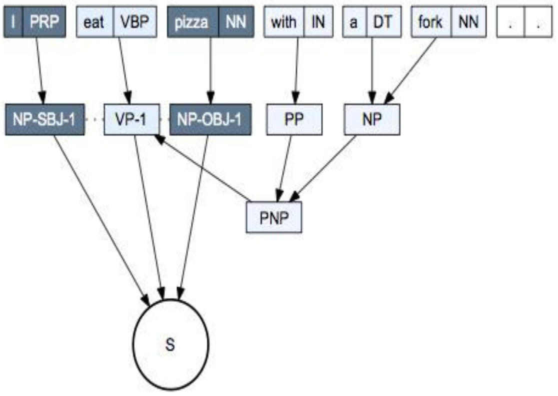

Fig: Shallow parsing Example

The syntactic parsing process is carved up into a set of classification problems, each of which can be separately learned using standard supervised machine learning methods. The MBSP modules include part of speech tagging, phrase labeling and grammatical relation finding for English. In applications such as information retrieval, question answering, and information extraction, where large volumes of (often ungrammatical) text have to be analyzed in an efficient and robust way, shallow parsing is useful.

2. Proposed System

We have proposed the architecture which combines the Methodology of NLIDB and KBIDB which we term as [6] NLKBIDB. Fig represents NLKBIDB architecture which takes Natural language query as an input and generates an output in tabular format if and only if generated SQL query is valid.

A. Natural Language Agent

Natural Language Agent takes natural language query as an input then it passes to the lexical Analyzer, Syntax Analyzer and Semantic Analyzer. Natural Language Agents sends analyzed tree structure to the SQL Generator if and only if Syntax of Natural language user query is valid. Working of Analyzer is as follows:

1. Lexical Analyzer

Natural language sentence provided by the user is given to Lexical Analyzer which will generate tokens from the sentence. It performs 3 operations:
1) Considering tokens delimited by space,
2) Uses list of stop words for not considering as a token,
3) Checking of the token based on domain.

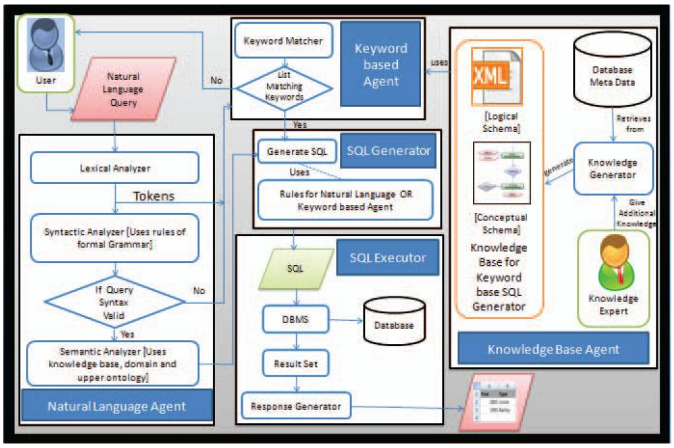

Fig: KNLIDB

2. Syntactic Analyzer

Syntactic analyzer parses the natural language sentence to generate the parse tree. If Syntax of the Sentence is not valid then it sends incorrect natural language query to the Keyword based Agent else SQL Generator will receive the parse tree and apply rules to generate SQL.

3. Semantic Analyzer

Semantic Analyzer analyzes the terminal nodes of the parse tree to generate proper keyword which belongs to database using generated knowledge base with Domain and Upper ontology.

B. Keyword Based Agent

Keyword Based agent process the lexicons passed by the Syntactic Analyzer. It applies mapping of tokens with the generated Knowledge base. If token is found in knowledge base then it will pass pointer to the SQL generator else It notify the user to reform the natural language query.

C. SQL Generator

SQL Generator generates SQL against two types of Rules.
1) Rules for Natural Language: If Syntax of the Query is correct; it will match the general rules of Noun, Verb, adjectives etc… to map with Select query of SQL. And
2) Rules for Keyword based interface: For Grammatical incorrect syntax, it matches the keyword with generated XML knowledge base to form the SQL Query.

D. SQL Executor

SQL Executor takes SQL as an input and generates the Result set using DBMS. It generates the output in tabular format if data is available else displays “No Data Found”.

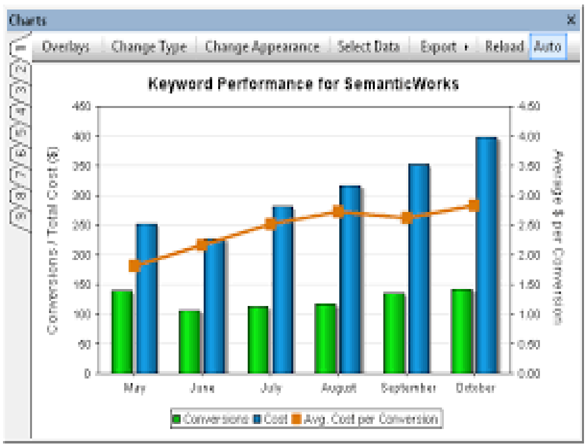

Chart -1: KNLIDB

The simple keyword-based query interface has to a great extent contributed to the wide acceptance of the Internet and its proliferation of user-contributed contents. The interface allows users to query vast collections of information freely, and hence improves the usability of the technology. Over the past two decades, database systems have made great advances in terms of performance, scalability and fault tolerance. They can now process a huge number of concurrent complex queries efficiently over enormous and diverse data sets.

3. CONCLUSIONS

Research is done from the last few decades on Natural Language Interfaces. With the advancement in hardware processing power, many NLIDBs mentioned in historical background got promising results. Many NLIDB and KBIDB mentioned in literature apply various techniques to generate SQL. Our combined approach NLKBIDB provides solution for syntactically correct and incorrect query both by providing keyword based and natural language Interface to the database. Our experiment proves that incorrect natural language input query can be solved by keyword based interface which derives Meta data as a knowledge base from database and updated by domain expert. In future we will extend the implementation for Syntactical Analyzer using link Grammar API, Semantic Analyzer, Extension of rules to cover different types of input query and formation of complex SQL queries to improve the efficiency and accuracy of proposed architecture.

ACKNOWLEDGEMENT

We feel great pleasure in presenting this partial report on Natural Language Interface to Database. We take this opportunity to thank all those who have contributed in successful completion of this report.

We are also thankful to Dr. V. H. Patil, Head of Department, Computer Engineering, MCERC, Nashik, for her timely suggestions and helpful guidance.

We are thankful to our guide Prof Swati.A.Bhavsar, for her guidance and encouragement in this work. Her expert suggestions and scholarly feedback had greatly enhanced the effectiveness of this work.

REFERENCES

[1] D. S. M. Mrs. Neelu Nihalani, “A Natural Language Interface for Database: A brief review,” ijcsi International journal of computer science, pp. 1694-0814, 2011.

[2] Bertino, B. Catania, G.P. Zarri, “Intelligent Database Systems”, Reading, Addsion Wesley Professional, 2001.

[3] Kamran Parsaye, Mark Chignell, Setrag Khoshafian and Harry Wong, “Intelligent databases-object-oriented, deductive hypermedia technologies”, New York, John Wiley& Sons, 1989.

[4] Androutsopoulos, G.D. Ritchie, and P. Thanisch, Natural Language Interfaces to Databases - An Introduction, Journal of Natural Language Engineering 1 Part 1 (1995), 29–81

[5] International Journal of Advanced Research in Computer and Communication Engineering Vol. 3, Issue 2, Jadhav Sneha ISSN (Online): 2278-1021 ISSN (Print): 2319-5940 February 2014

[6] International Conference on Advances in Computing, Communications and Informatics (ICACCI) NLKBIDB - Natural Language and Keyword Based Interface to Database Axita Shah AESICS, Ahmedabad University Ahmedabad, India.

[7] N. K. &. G.A.Patil, “Natural Language Database Interface for Selection of Data Using Grammar and Parsing,” in World Academy of science, Kolhapur, 2009.

[8] Y. S. Hu Li, “A Word Net-Based Natural Language Interface to Relational Databases,” in IEEE, Wuhan, China, 2010.