IOSR Journal of Computer Engineering (IOSR-JCE)
e-ISSN: 2278-0661, p- ISSN: 2278-8727Volume 16, Issue 3, Ver. IV (May-Jun. 2014), PP 99-103
www.iosrjournals.org

Natural Language Query Processing on Dynamic Databases Using Semantic Grammar

Sefali koli, Anjali Sarvade, Ashwini Waykar, Shradha Modak, R.A.Khan

Abstract: *Ease and effectiveness of Information Retrieval from Structured Database through Natural Language provides high utility value. The main purpose of Natural Language Query Processing is that an English sentence will be interpreted by the computer and appropriate action taken. It includes correct interpretation, disambiguation and context resolution in natural language query processing.*
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. Input of system is an English statement that will be converted into equivalent SQL query and SQL query will be fired on related database to get final output. Our system provides intelligence to the natural language processing in terms of lexicon processing, query formulation and automatic generation of SQL code which can be easily modified and reuse.
Keywords: *Natural language Processing, Semantic Grammar, POS.*

I. Introduction

Natural language processing is becoming one of the most active areas in Human-computer Interaction. It is a branch of AI (Artificial Intelligence) which includes Information Retrieval, Machine Translation and Language Analysis. The goal of

Modern Education Society's College of Engineering, Pune-01.

NLP is to enable communication between people and computers without resorting to memorization of complex commands and queries. In other words, NLP is a technique which can make the computer understand the languages naturally used by humans. While natural language may be simple for people to learn and use, it has proved to be the hardest for a computer to master.
The development and availability of efficient and appropriate search functions are still a challenge in the field of database and information systems. With a natural language interface users are able to communicate with the system as they are interacting with another user.

II. Literature Survey

In earlier seventies, Woods developed system called - LUNAR that answered questions about rock samples brought back from the moon. To accomplish its function, the Lunar system uses two databases; one for chemical analysis and other for literature references. The Lunar system uses an Augmented Transition Network (ATN) parser and procedural semantics. Here the system was not subject to intensive use due to limitation in linguistic capabilities. [3] In early eighties, David Warren developed natural language interface to a geographical database- CHAT-80: It gives information about for facts like oceans, major seas, major rivers and major cities. The system uses semantic grammar techniques and it is implemented in Prolog language. Major problem in this system is it is limited to only 150 countries and also implemented to specific database application only.

In early nineties, Kartz developed open domain question answering system - START: (SynTactic Analysis using Reversible Transformations). It ask question in English about MIT AI laboratory, geography and other topics. It uses semantics parsing. It handles all variety of media including text, diagrams, images, audio and video clips, data sets, web pages etc. The major problem in this system it only accepts questions related to its given domain like Geography, Science and Reference, Arts and Entertainment, History and Culture. It uses predefine template like subject-verb-object. Zhiping Zeng developed open-domain question answering system namely ANSWERBUS. It is asked on sentence level information retrieval which gives you a list of answers, each of which is a hyperlink to the source page. [2] It accepts users' natural-language questions in English, German, French, Spanish, Italian and Portuguese and extracts possible answers from the Web. Ruwanpura developed - SQ-HAL: It uses top down parser methodology.

The major drawback of this system is if appropriate database driver not installed, program will not work and moreover system does not support Microsoft Access database. The database table names and column names have to be valid English words. Here user has to manually enter the synonym for table names and

column names. It is not capable of determining relationship between tables. English Query (EQ) - It's a part of Microsoft SQL Server. : It uses relationship based on verbs. It consists of runtime engine and authorizing tools. It is developed using SQL project wizard. It uses context free Grammar rather than programming language. It includes database objects, semantic objects and relationships between them. The major problem of this system is it doesn't rely on any template; also it is available for Win32 platform only. It supports only database that have OLE database services like Oracle and Microsoft Access. The study reported above represents the direction of research in open domain Question Answering System and Natural language interface for Database.

Our system provides intelligence to the naturallanguage processing in terms of lexicon processing, query formulation and automatic generation of SQL code which can be easily modify and reuse again.

III. System Description

The system description is as follows: Suppose we consider a database and we have placed certain tables, which are properly normalized, now if the user wishes to access the data from the table, he/she has to be technically proficient in the SQL language to write a query for the database. [4] Our system eliminates this part and enables the end user to access the tables in his/her language. Let us consider an example: Suppose if we want to view information of a particular employee from EMP table then we are supposed to use the following query: SELECT SALARY FROM EMP WHERE name ='ABC';

But a person, who doesn't know SQL, will not be able to access the database unless he/she knows the syntax and semantics of firing a query to the database. But using NLP, this task of accessing the database will be much simpler. So the above query will be rewritten using NLP as: What is the salary of employee "ABC".

IV. Scope Of The System

The scope of the proposed system is as follows:

- To work with any RDBMS one should know the syntax of the commands of that particular database software (Microsoft SQL, Oracle, etc.).
- The Natural language processing is done for English language i.e. the input statements have to be in English.
- Input from the user is accepted in the form of questions only (wh- form like what, who, where, etc).
- Ambiguity among the words will be taken care of while processing the natural language.
- All the names in the input natural language statement have to be in double quotes.

-The database can be of two types:

✓ Static database

✓ Dynamic database

Static database

-A limited Data Dictionary is used where all possible worlds related to a particular system will be included.

- Data Dictionary used will be: - EMP, DEPT and PROJECT

Dynamic database

-User can dynamically create the database and fire the query.

-The mapping of English statement to SQL query is done at runtime.

-The Data Dictionary of the system will be regularly updated with words that are specific to the particular system.

V. System Architecture

Generally NLP has following steps:-

✓ Morphological Analysis:

Individual words are analyzed into their components and non word tokens such as punctuation are separated from the words.[1]

✓ Syntactic Analysis:

Linear sequences of words are transformed into structures that show how the words relate to each other.

✓ Semantic Analysis:

The structures created by the syntactic analyzer are assigned meanings.

✓ Discourse integration:

The meaning of an individual sentence may depend on the sentences that precede it and may influence the meanings of the sentences that follow it.

✓ Pragmatic Analysis: The structure representing what was said is reinterpreted to determine what was actually meant.

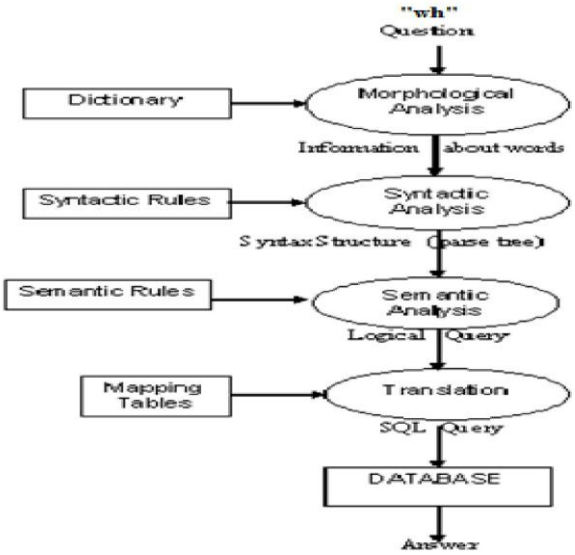

Fig1. Stages in Natural language interpretation process

Our proposed system includes the following modules:

✓ GUI:

Designing the front end or the user interface where the user will enter a query in Natural Language. Parsing: Derives the Semantics of the Natural Language Query given by the user and parses it in its technical form.

✓ Query Generation:

After the successful parsing of the statement given by the user, the system generates a query against the user statement in SQL and further gives it to the back end database.

✓ Data Collection:

This module collects the output of the SQL statement and places it in the User Interface Screen in order to display the result.

VI. System Implementation

The natural language input is first analyzed by lexicon analysis (document index, lexicon rules and spell check) and determine the domain of database, after which it directly goes to the history log. [1] If same question is repeated by the user, then it executes the structured query directly and displays output to the user else it forwards the question to the parser as shown in fig.2. The parser will syntactically parse the question by constructing a set of rules and generate POS tagging which can be used to identify proper noun, adjective, verb, etc and also generate a parse tree, which can be used by semantic interpreter to transform into intermediate query using semantic rules.

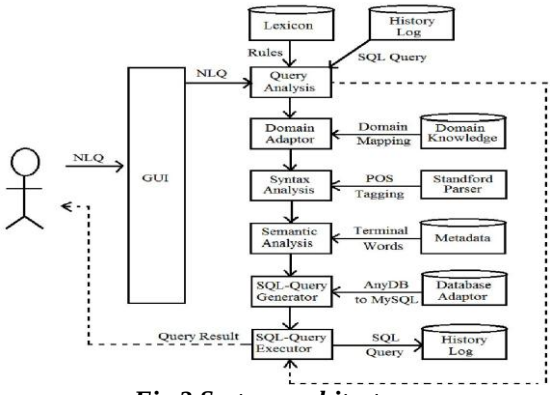

Fig.2 System architecture

The intermediate logical query generated by semantic interpreter, does not specify how to search the database in order to retrieve specific information. In order to retrieve desired output required by the user, the intermediate representation of query can be converted to some database query language.

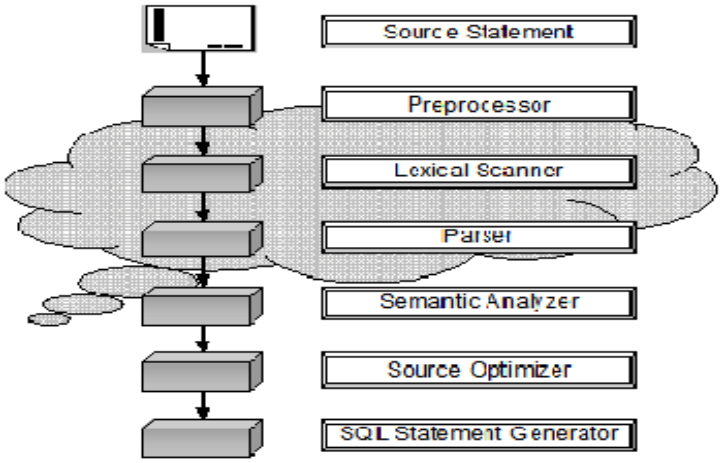

Fig.3 NLQ Parsing

The approach used here is to link logic predicate to SQL SELECT statement, nouns are mapped to attribute name and special nouns are mapped to entity variables.[4] Our goal is to design an interface for generating queries from natural language statements/questions. We are also designing a parser for the natural language statements, which will parse the input statement, generate the technical query and fire it to the end- database as shown in fig.3. Our application will understand the exact meaning which the end user wants; it will generates a query and give it to the interface. The interface further processes the query and searches the database. The database gives the result to the system and the result is displayed to the user.

VII. System Design

The graphical User Interface is as shown below. The user has to first login and then connect to the database. A database setting is required to access the database after getting the information about host name, database name, user name and the password. Once the database is setup, English Query can translate very complex English queries to SQL with the capability of searching multiple tables and multiple fields. [3]

Steps followed to get the result:-

- Type the natural language query in the dialogue box given
-.Click on "Proceed" button- The system will ask the user for the expected meaning. In case of ambiguities the user has to select the desired query.
- Click on "Generate SQL" button.
- The system will generate SQL query.
- Click on "Run Query".
- The result of the query will be displayed in the Output box.

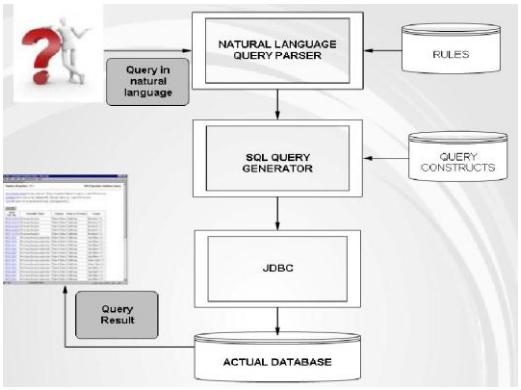

Fig.4 System Overview diagram

VIII. Overview Of Process

Firstly, the English sentence is parsed by the semantic grammar, then a postprocessor matches table and attributes names and joins up tables if the query involves more than one table .Next post-processor constructs the resulting SQL query and output it. [1] Information regarding the mapping of database tables and their attribute names, to the real world entities needs to be known. Finally we need to know the table level constrains on different tables. Attributes that are foreign keys would not be considered in this phase, since they link tables to other tables, and the appropriate phrases would have been covered by the table-table relationships. To translate natural language queries, system needs to have a parser, which contains all these grammar rules. The parser also has to know the table names, column names, any relationships between tables and related words for the table and column names. Only the most common grammar is included in the parser

IX. Algorithm

- Tokenization (scanning)
- Split the Query in tokens
- Give order number to each token identified
- Split Query and extract patterns
- Look for sentence connectors/criteria words
- Break Query on the basis of connector/criteria tokens
- Use criteria tokens to specify condition in query
- Find attributes and values after criteria token
- Map value for identified attribute and corresponding table
- Replace synonyms with proper attribute names
- Get intermediate form of Query
- Transform it into SQL

X. Future Enhancement

More grammar can be added to the parser to increase its effectiveness. Adding synonyms is another suggestion, which could help automating the related words for table and column names. So that the user can input appropriate words with table or column names and also remove unwanted words. So far, this system considers selection and a few simple aggregations. The next step is, to accommodate more complex queries. This system is currently capable of handling simple queries with standard join conditions. Because not all forms of SQL queries are supported, further development would be required before the system can be used within NLQP.

In future we can develop an android based application where a user can type or speak his query and appropriate results will be displayed on his screen.

XI. Conclusion

Natural Language Processing can bring powerful enhancements to virtually any computer program interface. This system is currently capable of handling simple queries with standard join conditions. Because not all forms of SQL queries are supported, further development would be required before the system can be used within NLDBI. Alternatives for integrating a database NLP component into the NLDBI were considered and assessed.

References

[1]. Gauri Rao, Chanchal Agarwal, Snehal Chaudhry, Nikita Kulkarni, Dr. S.H. Patil, "Natural Language Query Processing using Semantic Grammar", (IJCSE) International Journal on Computer Science and Engineering Vol. 02, No. 02, 2010, 219-223.

[2]. Valentin Tablan, Danica Damljanovic, and Kalina Bontcheva ,"A Natural Language Query Interface to Structured Information", Department of Computer Science University of SheieldRegent Court.

[3]. Mrs.Vidya Dhamdhere, Nijesh Hirpara, Kalpesh Surana, Karishma Gangwani, Chirayu Bootley, "A Natural Language Query Processor for Database Interface", Int.J.Computer Technology & Applications,Vol 3 (1), 378-382.

[4]. Gauri Rao, S.H.Patil,"Natural Language Query Processing based on Probabilistic Context Free Grammar" Department of Computer Engineering Bharati Vidyapeeth College of Engineering, Pune , India.

[5]. David Taniar, Hui Yee Khaw a, Haorianto Cokrowijoyo Tjioe a, Eric Pardede ,"The use of Hints in SQL-Nested query optimization",Clayton School of Information Technology, Monash University Clayton, Vic. 3800, Australia Department of Computer Science& Computer Engineering, La Trobe University, Bundoora, Vic. 3083, Australia