ISSN (Online) : 2278-1021
ISSN (Print) : 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 3, Issue 2, February 2014

Natural Language to Database Interface

Jadhav Sneha[1], Raut Shubhangi[2], A.S.Zore[3]

Student, Information Technology, Marathwada Mitra Mandal's Institute Of Technology, Pune, India[1,2]

Professor, Information Technology, Marathwada Mitra Mandal's Institute Of Technology, Pune, India[3]

Abstract: In our day-to-day life we always have to deal with information and information is playing an important role in our lives. The main source of information is database. Today we all use database technology which are having foremost impact on the growing use of computer almost today all IT application stores and retrieves the information from the database. To retrieve the information from database one has to know the structure of database languages like SQL. But however, not everyone is able to write SQL queries since they may not have the knowledge of database. And this has lead to the developing such a system where non-expert users compose their questions in their natural language and obtain the results in the form of tables. So instead of working with the SQL one can use to query relational databases in their natural language. So here new idea is provoked to develop new type of processing called natural language to database interface. Natural language to database interface enhances the users in performing flexible querying in database. This paper introduces to Natural Language to Database interface where information is extracted from the database just by entering query in Natural Language

Keywords*:* Natural Language processing, SQl, Information Extraction,Natural Language Interface to Database.

I. INTRODUCTION

In natural language to database interface, the user compose his or her query in natural language instead of structured database query. This is not a new area. A lot of research is going on since a long time from the sixties to till now. Now, the natural language seems to be an alternative interface for getting structured information from database. Writing questions in English or any other natural language is much easier for a non-expert user than a traditional graphical user interface for database access. The traditional methods which are mostly used for database access is complicated and requires time consuming navigation. The natural language to database interface shifts a user's burden of learning a structured database language to describe his or her need for information to the system. Now a days, there is a rising demands for non-expert users to work upon to querying relational database in a natural language encompassing linguistics variables and terms, instead of operating an the value of attributes.

The main objective of databases is to build the activities of data storage, data processing, and data retrieval etc. all of which are related with data administration in information systems. To retrieve information from a database, one needs to create a query in such way that the computer will understand and produce the desired output.

Retrieval of information from the databases requires the knowledge of databases language like the Structured Query Language (SQL).

The SQL norms are based on a Boolean construal of the queries. But not all the users are having knowledge of the Query language and not all the queries of user's seeking information from the database can be answered unambiguously by the querying system. It is due to the fact that not all the requirement's characteristics can be expressed by regular query languages. To simplify the complexity of SQL and to manipulate of data in databases for common people, many researchers have turned out to use natural language instead of SQL. The idea of using natural language instead of SQL has led to the development of new type of processing method called Natural Language Interface to Database systems. The NLDBI system is actually a branch of more inclusive method called Natural Language Processing. The main objective of NLP research is to create an easy and friendly environment to interact with computers in the sense that computer usage does not require any programming language skills to access the data; only natural language is required. Another shortcoming is that many NLP systems cover only a small domain of the English language questions. The system is general in nature given the appropriate database and knowledge base. This feature makes our system apparent. The huge amount of data are positioned in private personal computers. Mostly, the data is stored in some kind of databases. Data in database are usually managed by DBMS .The idea of using natural language instead of SQL has prompted the development of new type of processing method called Natural Language Interface to Database systems. NLDBI is a step towards the development of intelligent database systems to enhance the users in performing supple querying in databases.

II. EARLY SYSTEMS

Many systems relied on pattern matching to directly mapping the user input to the database. Formal List Processor (FLIP) is an early language for pattern-matching which was based on LISP structure works on the bases that if the input matches one of the patterns then the system is able to build a query for the database. In the pattern matching based systems, the database details were inter-mixed into the code, limited to specific databases and to the number and complexity of the patterns. Natural language processing was one of the approach, where the user interactively is allowed to question the stored data.

i. LUNAR system

The LUNAR system answers questions about samples of rocks brought back from the moon. This system was

ISSN (Online) : 2278-1021
ISSN (Print) : 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 3, Issue 2, February 2014

introduced in 1971. The system uses two databases; one for the chemical analysis and the other for literature references which helps it to achieve its functions. It uses an Augmented Transition Network (ATN) parser and Woods' Procedural Semantics. LUNAR system performance was somewhat remarkable; it handles 78% of requests without any errors and this ratio increased to 90% when dictionary errors were corrected. The system was not put to rigorous use because of its linguistic limitations.

ii. LADDER

The LADDER system was designed to give information about US Navy ships. LADDER system uses semantic grammar to parse questions to query a distributed database. It uses semantic grammars technique that interleaves syntactic and semantic processing. By parsing the input and mapping the parse tree to a database query the question answering is done. The first component of the system is for Informal Natural Language Access to Navy Data (INLAND). This component accepts questions in a natural language and generates a query to the database.

The queries from the INLAND are directed to the Intelligent Data Access (IDA). This component which is the second component of LADDER builds a fragment of a query to IDA for each lower level syntactic unit in the English Language. Input query and these fragments are then combined to higher level syntactic units to be recognized. The combined fragments are sent as a command to IDA at the sentence level. IDA would compose an answer that is relevant to the user's original query in addition to planning the correct sequence of file queries. The third component of the LADDER system is for File Access Manager (FAM). The FAM finds the location of the generic files and handles the access to them in the distributed database. LISP was used to implement the LADDER system

III. METHODS FOR NATURAL LANGUAGE PROCESSING INTERFACE

Natural language is the area of interest from computational perspective due to the hidden uncertainty that language possesses .Many researchers apply different techniques to deal with language. Next few sub-sections describe miscellaneous strategies that are used to process language for a range of purposes.

a. Symbolic Approach (Rule Based Approach)

Natural Language Processing is a robustly symbolic activity. Words are symbols that set for objects and concepts in real worlds, and they follow the well meticulous grammar rules. Knowledge about language is clearly programmed in rules. Language is analysed at various levels to obtain information. On this obtained information several rules are applied to achieve linguistic functionality. As Human Language include rule-based reasoning, and it is supported well by representational processing. The rules are formed for every level of linguistic analysis.

b. Empirical Approach (Corpus Based Approach)

This approach is based on statistical analysis and data driven analysis of raw data. And which is in the form of a collection of machine readable text. The approach has been approximately since NLP began in the early 1950s. The empirical NLP replaced rule-based NLP in the last decade. Corpora are used as a basis of information about language and many techniques have discovered to enable the analysis of corpus data. Syntactic analysis can be achieved on the basis of statistical probabilities estimated from a training corpus. Lexical ambiguities can be resolved by considering the possibility of one or another interpretation on the basis of context. This approach shows a positive result. Several different symbolic and statistical methods have been in use,
but they are used to generate a larger information mining system.

IV. PROPOSED SYSTEM

a. Problem Statement/System Architecture

Our proposed system NLDBI first transform the natural language question into transitional logical query, expressed in some internal meaning illustration language. The intermediate logical query expresses the meaning of the user's question in terms of high level world concepts. The logical query is then translated into the database's query language, and evaluated against the database.

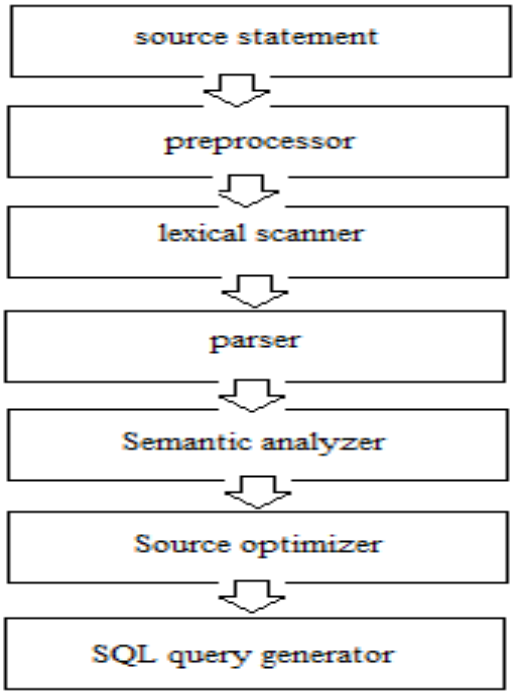

Fig 1: System Architecture

b. 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.

c. 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

ISSN (Online) : 2278-1021
ISSN (Print) : 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 3, Issue 2, February 2014

1) Tokenization/Lexical Scanner

The tokenizer's input is natural language question and its output is set of all possible complete tokenization of the questions. The tokenizer proceeds by stemming each word in the question. For each potential token the tokenizer checks whether the other words in the token are also present in the question. For example word salary contained in the token matching several database attributes such as: (employee. salary, manager. Salary)

It scans the given input sentence character by Character.

E.g. Who is the president of America?

By scanning tokens, it retrieves the set of database for matching tokens.

2) Text Normalization

It matches or maps' the words or tokens in the input to those present in the dictionary.

3) Tagging

Tagging is the process of marking up the words to a particular part of speech/syntactic category.

Example:

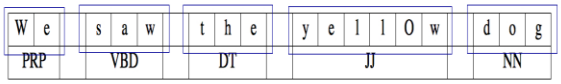

Fig.2 tagging of input

d. 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**.**

A sentence of the language is defined as a sequence of words such that the links connecting the words satisfy the following properties:

1) The links do not cross

2) The words form a connected graph,

3) The links satisfy the linking requirements of each word in the sentence.

The output of the parser, called a linkage, shows the dependencies between pairs of words in the sentence.

Parser is also capable of outputting constituent trees. A constituent tree is a syntactic tree of a sentence with the nodes represented by part-of-speech tags and words of the sentences in the leaf nodes. For instance, the corresponding constituent tree for the above sentence is illustrated in Fig. 2

E.g. I prefer morning flight.

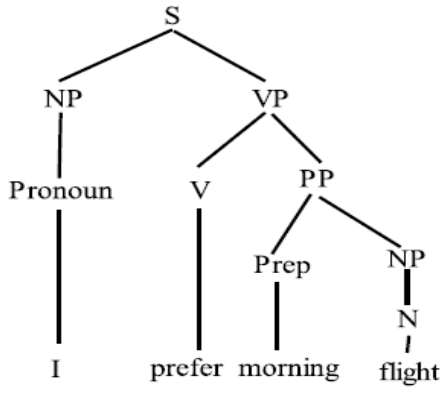

Fig.3 Parse Tree for a sentence

Figure out the meaning of linguistic input (construct meaning representations) process language to produce common-sense knowledge about the world (extract data and construct models of the world)

e. Semantic Analyzer

Figure out the meaning of linguistic input (construct meaning representations) process language to produce common-sense knowledge about the world (extract data and construct models of the world)

Often the task of a language processor is to analyse a sentence in a language like English and produce an expression in some formal notation which, as far as the computer system is concerned, concisely expresses the semantics of the sentence. An interface to a database might, for example, require a language processor to convert sentences in English or German into SQL queries. Semantic analysis is the term given to the production of this formalized semantic representation.

In order to carry out semantic analysis the lexicon must be expanded to include semantic definitions for each word it contains and the grammar must be extended to specify how the semantics of any phrase are formed from the semantics of its component parts. For example the grammar rule above

Verb Phrase → Verb, Noun Phrase states how the syntactic group called *Verb Phrase* is formed from other syntactic groups but says nothing about the semantics of any resulting *Verb Phrase*. Using a simplified form of logic the grammar and lexicon can be expanded to capture some semantic information. This is illustrated in the following example.

Syntactic rule
Sentence→noun phrase ,verb phrase Verb phrase→verb, noun phrase Noun phrase→article, noun Noun phrase→article, adjective, noun phrase

There are two semantic interpretations for this sentence. Using a form of logic for the semantics:

1. put the apple which is currently in the basket onto the shelf (E_1 a : apple ÆE b : basket L inside(a, b)) L E_1 s : shelf fi puton(a, s)

2. put the apple into the basket which is currently on the shelf

E_1 a : apple L (E_1 b : basket ÆE_1 s : shelf L on(b, s)) fi Putin(a, b)

a. Source Optimizer:

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

b. *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.

An important characteristic of an IE system is its ability to extract high-quality outcome. It takes the database

ISSN (Online) : 2278-1021
ISSN (Print) : 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 3, Issue 2, February 2014

elements selected by the previous module and combines them into well-formed SQL query.
In relational query,
Select → WH words
Where → Conjunction of attributes
From → Relation name for the attribute in
E.g. Show all student info. Studying in BEIT. →
Select * from student where class="BEIT".

V. ADVANTAGES & DISADVANTAGES

a. Advantages

a) No need to learn SQL Language.

It is useful for that user who does not have knowledge of artificial communication language to query the database without learning the artificial language. Formal query languages like SQL are difficult to learn and master, at least by non-expert user.

b) Simple, reliable to use

An NLIDB system only requires a single input, a form-based may requires multiple inputs depending on the need of that form. In the case of a query language, a question may need to be expressed using multiple statements which contain one or more sub- queries with some joint operations.

c) Better for various Questions

There are some kind of questions that can be easily expressed in natural language, but that seem difficult to express using graphical or form-based interfaces For example, "Which department has no developers?" or "Which company supplies sales department?" can be easily expressed in natural language, but they are difficult to express in graphical or form-based interfaces. They can be expressed using the query language but again it would require large complex queries which can be only written by the computer experts.

d) Fault tolerance

In a computer query language the syntax and the rules of the language must be followed and any errors will cause the input automatically be discarded by the system while most of NLIDB systems grant some tolerances to trivial grammatical errors.

e) Easy to Use for Multiple Database Tables

Queries that involve multiple database tables like "list the address of the employees who got bonus greater than 10000 rupees for their work", are difficult to form in graphical user interface as compared to natural language interface.

b. Limitation

a) Forged expectations

People may assume that the system is intelligent and they expect system's ability to process a natural language and produces correct output .Therefore rather than asking accurate questions from a database, they may ask questions that may have complicated ideas, certain judgments, explanation capabilities, etc. which an NLIDB system cannot give response.

VI. CONCLUSION

NLP is relatively recent area of research and application. We have described a theoretical as well as practical approach to the problem producing a reliable NLI to database.

Now-a-days such interfaces are increasingly important on websites, particularly as people access information more from cell phones and PDA's and other small screen devices where GUI is less appealing.

In our novel framework we store an intermediate processed data introducing new knowledge can be issued with simple SQL insert statements on top of the processed data, on the other hand existing extraction framework do not provide the capabilities of managing intermediate processed data. Our framework is most suitable for performing extraction on text corpus written in natural sentences. Our extraction approach saves much more time.

REFERENCES

[1] Incremental Information Extraction Using Relational Databases [IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, VOL. 24, NO. 1, JANUARY 2012] Luis Tari, Student Member, IEEE Computer Society, Phan Huy Tu,Jo¨ rg Hakenberg, Yi Chen, Member, IEEE Computer Society,Tran Cao Son, Graciela Gonzalez, and Chitta Baral

[2] Natural language Interface for Database: A Brief review IJCSI International Journal of Computer Science Issues, Vol. 8, Issue 2, March 2011, www.IJCSI.org

[3] Natural Language Web Interface for Database Proceedings of the Third International Symposium, SEUSL: 6-7 July 2013, Olivia, Sri Lanka

[4] GenerIE: Information Extraction Using Database Queries.(Luis Tari1, Phan Huy Tu1, Jorge Hakenberg1)Department of Computer Science and Engineering, Arizona State University Tempe, AZ 85287, USA

[5] Efficient Information Extraction over Evolving Text Data, Jun Yang, Raghu Ramakrishnan Duke University, Yahoo! Research

BIOGRAPHY

Prof. A.S.Zore Is currently working as an Assistant Professor in Marathwada Mitra Mandal's Institute Of Technology, Pune, Maharashtra, India. His research interests are Computer Networks, Software Engineering, and Data Mining etc. **Prof.A.S.Zore** has completed M.E. (Comp.Networks). His research interests are Data Mining, Network Security etc.