Volume 5, Issue 4, 2015 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

A Survey of Natural Language Query Builder Interface to Database

Deepshikha[1], **Kiran Devi**[2]
[1]Assistant Professor, [2]M.Tech Student
Department of Computer Science and Engineering,
Kurukshetra Institute of Technology &Management, Haryana, India

*Abstract:— **Natural language processing is an area of artificial intelligence that defines a set of methods and techniques used to automate the translation process between computers and humans or mediate the human-machine communicationThe necessity for Hindi Language interface has become growingly error free as native people are using databases forstoring the data. Large number of government applications like agriculture, weather forecasting, railways, legacy matters etc use database.It is very difficult for common user to access database using query languages as SQL. So a system is to be improved which make the user to access the database using their native language. This paper is an introduction to the natural language interface to databases (NLIDBS) and merits and demerits of NLIDB. Some exiting systems are then discussed followed by the discussion of the components of NLIDB. After that various techniques are discussed.***

Keywords---Hindi language interface to database,natural language interface to database, tokenizer, syntax analyser, parser,sql(structured query language),dbms (database management system).

I. INTRODUCTION

As we know in daily life we necessity information from time to time and source for that information is a database. To retrieve information from that database we require knowledge of database languages like SQL. For writing queries in SQL demand to have knowledge of formal language. But everyone is not experienced of writing those queries. So to overcome these difficulty researchers turns to use natural language (NL). i.e. English, Arabic, Spanish, Tamil etc. Instead of SQL.The design of using natural language instead of SQL prompted the development of a new type of processing method i.e. Natural Language Interface to Database. In which a user do not require to learn any other official language. They can type their query in their native language. So it has discarded the burden to learn SQL. By using this systems anyonecan retrive information from the database.

NLIDB

It is difficult for a user to access database without any kind of knowledge database language. In today’s world demand for inefficient users to query relational database is growing. Therefore the target of using natural language instead of SQL simulatesthe development of new type of processing method Natural Language Interface to Database. Here users have no need to learn any other formal language; they can typequery in their native language. Therefore it overcomes the problem to learn SQL.By using such systems user can collect information from the database. Also it may alter our thinking about the information in a database. Earlier, people are working with a form; their expectations depend massively on the capabilities of the form. Natural Language Interface to Database makes the entire approach more flexible. There are many applications that can take benefit of NLIDB system. In PDA and cell phone environments, the screen is not large as of a computer. Filling a form that has many fields can be tiresome. One may have to navigate through the screen, to scroll, to look up the scroll box values etc. Instead, with NLIDB, only work that needs to be done is to type the question similar to the SMS.

Component of NLIDB

To access database Computing developers have divided the problem of natural language into two components.

Linguisticcomponent:

Ithandles the natural language input, convert it into formal query and generate a natural language output as a result after execution.

Database Component:

It performsdatabase management functions. A lexiconcomprisenumber of tables to create a formal query that store natural language words and their corresponding checking to formal objects that will be used. These tables can have entries of table name,colum names, verbs, adverbs etc. query entered in natural language converted into a statement with the help of parser which tokenize the input. Then by mapping tokens into lexicon tables a formal query is formed. After that query is executed and the result in natural language is given to user.

II. LITERATURE REVIEW

Since the end of 1960 there has been a wide number of research paper indicatingthe theories and execution of NLIDBs. Asking question to databases in natural language is very convenient and simple method of data access especially for common users who do not understand complex database query language.

ExistingNLIDBsystems

Prototype for NLIDB had appeared in late sixties and early seventies. Since then a number of systems have introduced. Here we discuss some of them .

LUNAR

LUNAR is a system which was developed in 1971. It answers questions about samples of rocks brought back from the moon. To accomplish its functionality the LUNAR system uses two databases; one for the chemical analysis and the other for literature references. The LUNAR system uses an Augmented Transition Network (ATN) parser and Woods' procedural Semantics.

LADDER

The LADDER system was designed as a natural language interface to a database of information about US Navy ships. The LADDER system uses semantic grammar to parse questions to query a distributed database. The system uses semantic grammars technique that interleaves syntactic and semantic processing. The question answering is done via parsing the input and mapping the parse tree to a database query. The system LADDER is depend on a three layered architecture. The first component of the system is for Informal Natural Language Access to Navy Data (INLAND), which receives questions in a natural language and develop a query to the database. The queries from the INLAND are conducted to the Intelligent Data Access (IDA), which is the second component of LADDER. The INLAND component 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. At the sentence level, the combined fragments are sent as a command to IDA. IDA would create an answer that is apposite 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). FAM find the location of the generic files and control the access to them in the distributed database. The system LADDER was implemented in LISP. At the time of the creation of the LADDER system was able to process a database that is equivalent to a relational database with 14 tables and 100 attributes.

Rendezvous System

Rendezvous system appeared in late seventies. In this system, end users could access databases through relatively unrestricted natural language. In this Codd‟s system, main consequence is placed on query description and fetching users in clarification conversation when there is difficulty in parsing user query.

Planes

This was appeared in late seventies (Programmed Language-based Enquiry System) at the University of Illinois Coordinated Science Laboratory. PLANES comprise an English language with the capability to understand and explicitly answer user requests. It carries out clarifying conversations with the user as well as answer ambigous or poorly defined questions. This task is being carried out using database depend upon information of the U.S. Navy 3-M (Maintenance and Material Management). It is a database of aircraft maintenance and flight data, although the ideas can be directly applied to other nonhierarchic record-based databases. Plane was developed in 1977 and also known as Philips Question Answering System, uses a syntactic parser which runs as a separate pass from the semantic passes. This system is mainly involved with problems of semantics and has three separate layers of semantic understanding. The layers are called "English Formal Language", "World Model Language", and "Data Base Language" and appear to correspond to the "external""conceptual", and "internal" views of data.

Chat-80

The system CHAT-80 is one of the most referenced NLP systems came in the eighties. The system was implemented in Prolog language. The CHAT-80 was an impressive, efficient and sophisticated system. The database of CHAT-80 comprises of facts (i.e. oceans, major seas, major rivers and major cities) about 150 countries of world and a small set of English language vocabulary that are enough for querying the database. The CHAT-80 system processes an English language query in three stages as show in figure.

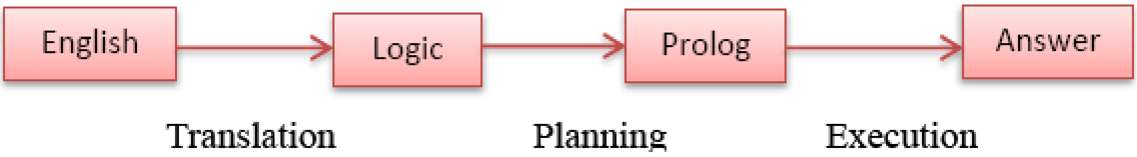

Figure: Working phases of CHAT-80

Team

It was developed in 1987 for portability issues. TEAM was to be designedeasily andconfigurable by database administrators with no knowledge of NLIDBs

Ask

This system named "ASK" was first made in 1983 and it also allowed users to let the system learn new words and concepts through interaction. Actually ASK was a information management system which provide its own built in database and the able to interact with multiple external databases and electronic mail programs. End user can access all applications on ASK by using Natural language query. User enters his query in English and this system generates suitable query to appropriate underlying system.

Datalog

DATALOG is a database system wchich query in cascade ATN grammar by providing different representation schemes for linguistic information and application domain information, general world information and application domain knowledge. Datalog achieve a high degree of extendability and movability.

Precise

Precise is a system made at the University of Washington by Ana-Maria Popescu, Alex Armanasu, Oren Etzioni, David Ko, and Alexander Yates . The database is in form of table using SQL as the query language. It initiates the concept of semantically tractable sentences that can be translated into a unique semantic interpretation by analyzing some lexicons and semantic constraints. Precise was dependon two database domains. The first one is the ATIS domain, which comprises of spoken questions about air travel, their written forms, and their correct translations in SQL query language. In ATIS domain, 95.8% of questions were tractable semantically. The second domain is the GEOQUERY domain. This domain consist information about U.S. Geography, 77.5% of the questions in GEOQUERY are semantically tractable. After executing these questions PRECISE give 100% accuracy. The strength of PRECISE is based on the capacity to match keywords in a sentence to the corresponding database structures. This task is done in two stages, first by reducing the possibilities using Maxflow algorithm and second by examining the syntactic structure of a sentence. Therefore PRECISE iscapable to perform semantically manageable questions.

JANUS(1989)

It had abilities to interface with severalprimitive systems (databases, expert systems,graphics devicesetc). All the premitivesystems could participate in the rating of a natural language request without the user ever becoming aware of the heterogeneity of theoverall system. JANUS is also one of thesystems that support temporal questions.

GINLIDB

GINLIDB stands for Generic Interactive Natural Language Interface to Databases.GINLIDB system consists of two major components:

1. Linguistic handling component
2. SQL constructing component

The first component controls the natural language query faultlessness as far as grammatical structure and the successful transformation to SQL query.The second component opens a connection to the database , generates the appropriate sql statement and executes the generated SQL statement and returns the result to the user. The GINLIDB parser is purposed with two stages of grammars:

lexical and
syntactic.

The second stage is asyntactic analysis depend on Augmented Transition Network (ATN) which checksif the tokens' structure is in correct grammatical form. Then it is handled viathe parser conforming to a Context-FreeGrammar (CFG).

Wasp

It stands for Word Alignment-based Semantic Parsing. It was made at the University of Texas, Austin by Yuk Wah Wong. Wasp system was designed for achieving the target of establishinga comprehensive, formal, symbolic, and significant representation of a natural language sentence that can also be executed to the NLIDB domain. Prolog was used as the

Formal query language. WASP studys to construct a semantic parser given a collection of natural language sentences explain with their right formal query languages. It has no previous knowledge requirement of the syntax, because the whole learning process is done using statistical machine translation techniques

III. TECHNIQUES USED FOR DEVELOP IN GNLIDBS

Pattern-Matching

In the pattern matching based systems, the database specific were inter-mixed into the code, restricted to specific databases, to the number and complication of the patterns. The main good of the pattern-matching approach is its simplicity. In such systems no detailed parsing and interpretation modules are required, and the systems are also easy to implement.

Syntax-BasedSystems

In syntax-based systems the users input is parsed and the resulting parse tree is directly depicted to an expression in database query language. Syntax-based systems conduct a grammar that explain the possible syntactic structures of the user's questions. The main advantage of using syntax based approaches is that they deliver detailed information about the structure of a sentence. A parse tree contain a collection of information about the sentence structure; starting from a single word and its part of speech, how words can be clustered together to form a phrase, how phrases can be assembled together to form more complex phrases, until a complete sentence is built. By having this information, we can map the semantic meanings to certain production rules.

SemanticGrammarSystems

The basic design of a semantic grammar system is to make simpler the parse tree as much as possible, by removing unneeded nodes or combining some of the nodes at the same time. Based on this design, the semantic grammar system reflect the semantic representation without having complicated parse tree structures. The main issue of semantic grammar approach is that it needs some prior knowledge of the elements in the domain, therefore making it hard to port to other areas. In addition, a parse tree in a semantic grammar system has specific structures and unique node labels, which could hardly be useful for other applications.

IntermediateRepresentationLanguages

Intermediate Representation System was initiatebecause the problem of directly converting a sentence into general database query languages using syntax based approach. The idea is to translating a sentence into a logical query language first and then further converts this logical query language into a general database query language. In this process there can be more than one intermediate meaning representation language.

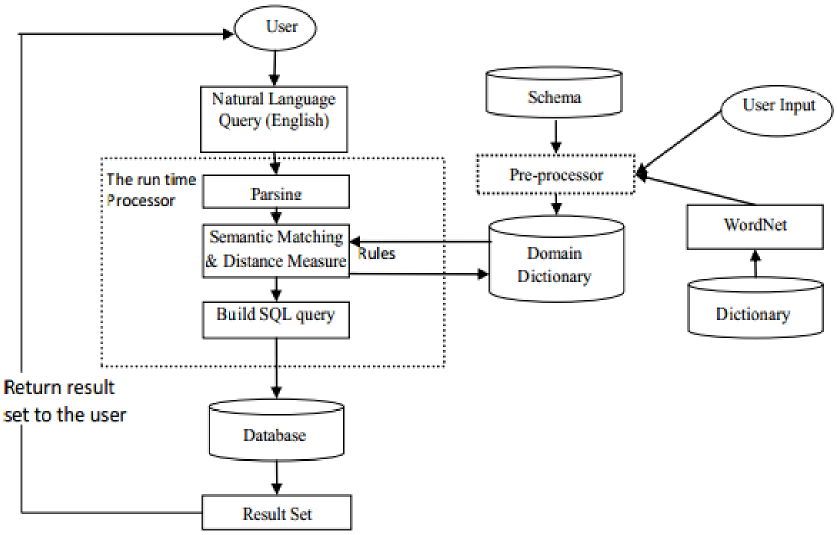

Figure 1 Architecture of our NLIDB

Figure: Architecture of NLIDB

ADVANTAGES OF NLIDB

Like other systems, NLIDB also have some merits as well as demerits. Advantages are following:

No need to know artificial language

As there are some languages which we called conventional query language are very hard to interpret. In NLIDB user can use their spoken language for querying the database, So no user has to read any kind of language for querying.

No need To know theOuter structure of data

To query in formal language one should be aware of location of the data where it is stored. But this is not required in NLIDB.

Easy to use

For taking data from NLIDB system we require a single input, while a form based may contain multiple input. In case of query language a question needs to be stated by using multiple statements which may consist one or more sub queries with some joint operations.

Discourse

Another merits of NLIDB creation in concern of natural language interface that support anaphoric and elliptical expression. NLIDB of this kind allow the use of brief underspecified questions where the sence of each query is assisted by the discourse context.

Easy to Use for More than one Database Tables

Queries that involve multiple database tables like “list thename of the farmers who lost crop worth more than 50000/- during flood” are hard to form in graphical user interface as compared to natural language interface.

Disadvantages of NLIDB

Many NLIDB systems have been developed so far for business purpose use but use of NLIDB system is not wide spread and it is not the primary choice for interfacing to database. The absence of acceptance is mainly due to the large numbers of disadvantages which are given below:

Linguistics coverage is not obvious:

At present all NLIDB systems can recognize some subsets of a natural language but it is quite hard to explain these subsets. Even some NLIDB systems can’t hold certain query belong to their own subsets. This is not the case of formal language like SQL. Because the formal language description is clear and give the corresponding answers of any statements that observe the given rules.

Linguistics vs. imaginary failure:

When NLIDB can’t interpret a question, it is often not clear to the user whether the refused question is out of system’s conceptual analysis or it is outside the system’s linguistic analysis. Thus user often try to give input by changing the phrase question referring to concepts the system does not knowbecause they think the problem is caused by the system’s restricted linguistic coverage

Inappropriate Medium:

It has been state that natural language is not an appropriate medium for communicating with a computer system. Natural language is claimed to be too wordy or too ambiguous for human-computer interaction. NLIDB users have to type long questions, while in form-based interfaces only fields have to be filled in, and in graphical interfaces most of the work can be done by mouse-clicking. In natural language interface user has to type full sentence with all the connecters (articles, prepositions, etc) but in graphical or form based interfaces it is not required .

Unrealistic expectations

Mostly people depend on NLIDB system’s capability to process a natural language query: they assume that the system is intelligent so it can comprehend facts. Therefore rather than asking precise questions from a database, they may be tempted to ask questions that involve complex ideas, certain judgments, reasoning capabilities, etc, which an NLIDB system cannot be relied upon.

IV. CONCLUSION

This system accepts query in Hindi language and convertthat query into SQL query by mapping words of Hindi language with the help of database maintained by mapping their corresponding words. Here we attempted to define two purposes: To introduce user to NLIDB by defining some issues and to demonstrating the current framework in this field by outlining the facilities and methods. The most important thing is that it’s user friendly. End user accepts this kind of system only if it is easy to use.Though several NLIDB systems have also been developed so far for commercial use but the use of NLIDB systems is not wide-spread and it is not a standard option for interfacing to a database. The main demerits of NLIDB is due to large number of deficiencies, One of which is to understand natural language.

REFERENCE

[1] Edith Buchholz, HeikoCyriaks, Antje Diisterhoft, HolgerMehlan, Bernhard Thalheim, “Applying a Natural Language Dialogue Tool for Designing Databases”.

[2] Paul S Jacobs, “Generation in a Natural Language Interface”, Division of Computer Science, Department of EECS, University of California, Berkeley, CA, USA

[3] Frank Meng and Wesley W Chu, “ Database Query Formation from Natural Language using semantic Modeling and Statistical Keyword Meaning Disambiguation”, Computer Science Department, University of California, Los Angeles, CA, 900095 USA.

[4] M. Dua, S. Kumar, “Hindi Language Graphical User Interface to Database Management System”, 12th International conference on Machine learning and Applications, Miami, USA 2013.

[5] A. Shingala, P. Virparia, “Enhancing the Relevance of Information Retrieval by Querying the Database in Natural form”, International Conference on Intelligent Systems and Signal Processing (ISSP), 2013 .

[6] N. Nihalani, S. Silakari, M. Motwani“Natural Language Interface for Database: A Brief Review”, International Journal of Computer Science Issues, vol. 8, Issue 2, March 2011.

[7] A. Shingala, P. Virparia ,” Enriching Document Features for Effective Information Retrieval using Natural Language Query Interface”, International Journal of IT, Engineering and Applied Science Research, ISSN: 2319-4413, 2012.

[8] B. Sujatha, S. ViswanadhaRaju and HumeraShaziya "A Survey of Natural Language Interface to Database Management System" International Journal of Science and Advance Technology", vol.2, no.6, June 2012.

[9] Abrahams P. W. et al. "The LISP 2 Programming Language and System", in proceeding of FJCC, No. 29, USA, 1996, pp. 661- 676.

[10] H. Jain, P. Bhatia, "Hindi language interface to database," 2011.

[11] A. Shingala, P. Virparia, "Enhancing the Relevance of Information Retrieval by Querying the Database in Natural form", International Conference on Intelligent Systems and Signal Processing (ISSP), 2013 .