'FUTURIZM-2014, (14TH & 15TH March 2014), 'P.D.E.As College of Engineering, Manjari[Bk]', 'IIP Cell,'Manjari Road, Hadapsar, Pune-412307. India.', 'Chief Editor:Dr.K.R.Harne', Editors: Prof R V Patil, Prof Niraja Jain

International Journal of Innovative Technology & Adaptive Management (IJITAM)
ISSN: 2347-3622, Volume-1, Issue-10 , July, 2014

Natural Language Processing based SQL query generation for retrieval of data from temporal / non-temporal database.

Satwik Kolhe <satkolhe@gmail.com>, Nitish Muley <nitish.muley@gmail.com>

Abstract: **Nowadays interaction with computer has become essential. Use of database for storing data has increased. Natural language processing techniques facilitate the interaction of novice users with the system in their native (English) language. Retrieving data from database needs prior knowledge of methods and techniques such as structured query language. The data retrieved from this database is based only on current data present in the database. Here we propose a system which will facilitate the interaction of novice user with the database in their native (English) language. The main aim of the system is to interpret the English query (i.e. English sentence), and generate appropriate results. The proposed system is focused to work on temporal database which will support past, present and future data. The proposed system accepts questions in English language and converts it into an SQL query which is then forwarded to database to retrieve the result of the question asked. This is achieved using natural language processing, context free grammar and parsing techniques.**

Keywords: Natural Language Processing, Context Free Grammar, Parsing.

1.0 Introduction

Natural language processing is becoming one of the most active areas in Human-computer Interaction. The goal of NLP is to enable communication between people and computers without resorting to memorization of complex commands and procedures. In other words, NLP is a technique which can make the computer understand the languages naturally used by humans, but not by artificial or man-made language such as a programming language. How to link those concepts together in a meaningful way. While natural language may be the easiest symbol system for people to learn and use, it has proved to be the hardest for a computer to master. Despite the challenges, natural language processing, or NLP, is widely regarded as a promising and critically important endeavor in the field of computer research. The general goal for most computational linguists is to imbue the computer with the ability to understand and generate natural language so that eventually people can address their computers through text as though they were addressing another person. The applications that will be possible when NLP capabilities are fully realized are impressive computers would be able to process natural language, translating languages accurately. and in real time, or extracting and summarizing information from a variety of data sources, depending on the users' requests.

2.0 Related Work

The very first attempts at NLP database interfaces are just as old as any other NLP research. In fact database NLP may be one of the most important successes in NLP since it began. 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. The success in this

International Journal of Innovative Technology & Adaptive Management (IJITAM)
ISSN: 2347-3622, Volume-1, Issue-6 , March, 2014

area is partly because of the real-world benefits that can come from database NLP systems, and partly because NLP works very well in a singledatabase domain. Databases usually provide small enough domains that ambiguity problems in natural language can be resolved successfully. LUNAR (Woods, 1973) involved a system that answered questions about rock samples brought back from the moon. Two databases were used, the chemical analyses and the literature references. The program used an Augmented Transition Network (ATN) parser and Woods' Procedural Semantics. The system was informally demonstrated at the Second Annual Lunar Science Conference in 1971. LIFER/LADDER was one of the first good database NLP systems. It was designed as a natural language interface to a database of information about US Navy ships. This system, as described in a paper by Hendrix (1978), used a semantic grammar to parse questions and query a distributed database. The LIFERILADDER system could only support simple one-table queries or multiple table queries with easy join conditions.

3.0 Components of the system

Computing scientists have divided the problem of natural language access to a database into two sub-components,

A. Linguistic Component: It is responsible for translating natural language input into a formal query and generating a natural language response based on the results from the database search.

B. Database Component : It performs traditional Database Management functions.

A lexicon is a table that is used to map the words of the natural input onto the formal objects (relation names, attribute names, etc.) of the database. Both parser and semantic interpreter make use of the lexicon. A natural language generator takes the formal response as its input, and inspects the parse tree in order to generate adequate natural language response. Natural language database systems make use of syntactic knowledge and knowledge about the actual database in order to properly relate natural language input to the structure and contents of that database. Syntactic knowledge usually resides in the linguistic component of the system, in particular in the syntax analyzer whereas knowledge about the actual database resides to some extent in the semantic data model used. Questions entered in natural language translated into a statement in a formal query language. Once the statement unambiguously formed, the query is processed by the database management system in order to produce the required data. These data then passed back to the natural language component where generation routines produce a surface language version of the response.

4.0 Various Approaches.

Natural language is the topic of interest from computational viewpoint due to the implicit ambiguity that language possesses. Several researchers applied different techniques to deal with language. Next few subsections describe diverse strategies that are used to process language for various purposes.

A. Symbolic Approach (Rule Based Approach): Natural Language Processing appears to be a strongly symbolic activity. Words are symbols that stand for objects and concepts in real worlds, and they are put together into sentences that obey well specified grammar rules. Hence for several decades Natural Language Processing research has been dominated by the symbolic approach (Miikkulainen, 1997).R. Akerkar and M. Joshi Knowledge about language is explicitly encoded in rules or other forms of representation. Language is analyzed at various levels to obtain information. On this obtained information certain rules are applied to achieve linguistic functionality. As Human Language capabilities include rule-base reasoning, it is supported well by symbolic processing. In symbolic processing rules are formed for every level of linguistic analysis. It tries to capturethe meaning of the language based on these rules.

B. Empirical Approach (Corpus Based Approach): Empirical approaches are based on statistical analysis as well as other data driven analysis, of raw data which is in the form of text corpora. A corpus is collections of machine readable text. The approach has been around since NLP began in the early 1950s. Only in the last 10 years or so empirical NLP has emerged as a major alternative to rationalist rule-based Natural Language Processing. Corpora are primarily used as a source of information about language and a number of techniques have emerged to enable the analysis of corpus data. Syntactic

International Journal of Innovative Technology & Adaptive Management (IJITAM)
ISSN: 2347-3622, Volume-1, Issue-6 , March, 2014

analysis can be achieved on the basis of statistical probabilities estimated from a training corpus. Lexical ambiguities can be resolved by considering the likelihood of one or another

interpretation on the basis of context. Recent research in computational linguistics indicates that empirical or corpus –based methods are currently the most promising approach to developing robust, efficient natural language processing (NLP) systems (Church, 1993; Charniak, 1993). These methods automate the acquisition of much of the complex knowledge required for NLP by training on suitably annotated natural language corpora, e.g. tree-banks of parsed sentences (Marcus, 1993).5.0 Architectures Most of the empirical NLP methods employ statistical techniques such as ngram models, hidden Markov models (HMMs), and probabilistic context free grammars (PCFGs).

5.0 Architectures

A. Pattern-matching systems : Some of the early NLIDBS relied on pattern-matching techniques to answer the user's questions. The main advantage of the pattern-matching approach is its simplicity: no elaborate parsing and interpretation modules (see later sections) are needed, and the systems are easy to implement. Also, pattern-matching systems often manage to come up with some reasonable answer, even if the input is out of the range of sentences the patterns were designed to handle. Returning to the example above, the second rule would allow the system to answer the question "Is it true that the capital of each country is Athens?", by listing the capital of each country, which can be considered as an indirect negative answer. Pattern-matching systems are not necessarily based on such simplistic techniques as the ones discussed above.

B. Syntax-based systems : In syntax-based systems the user's question is parsed (i.e. analysed syntactically), and the resulting parse tree is directly mapped to an expression in some database query language. A typical example of this approach is Lunar Syntax-based NLIDBS usually interface to applicationspecific database systems, that provide database query languages carefully designed to facilitate the mapping from the parse tree to the database query. It is usually difficult to devise mapping rules that will transform directly the parse tree into some expression in a real-life database query language.

C. Semantic grammar systems : In semantic grammar systems, the question-answering is still done by parsing the input and mapping the parse tree to a database query. The difference, in this case, is that the grammar's categories (i.e. The non-leaf nodes that will appear in the parse tree) do not necessarily correspond to syntactic concepts. Semantic grammars were introduced as an engineering methodology, which allows semantic knowledge to be easily included in the system. However, since semantic grammars contain hard-wired knowledge about a specific knowledge domain, systems based on this approach are very difficult to port to other knowledge domains a new semantic grammar has to be written whenever the NLIDB is configured for a new knowledge domain.

D. Intermediate representation languages : Most current NLIDBS first transform the natural language question into an intermediate logical query, expressed in some internal meaning representation language. The intermediate logical query expresses the meaning of the user's question in terms of high level world concepts, which are independent of the database structure. In the intermediate representation language approach, the system can be divided into two parts. One part starts from a sentence up to the generation of a logical query. The other part starts from a logical query until the generation of a database query. In the part one, The use of logic query languages makes it possible to add reasoning capabilities to the system by embedding the reasoning part inside a logic statement. In addition, because the logic query languages is independent from the database, it can be ported to different database query languages as well as to other domains, such as expert systems and operating systems.

6.0 Proposed system

Here the system is developed, for which the input should be given in English. The input may be question or be a simple sentence (like list, show, what, when etc.) This system is designed for temporal database in which we can get Past, Present as well as Future Data. This system also support for validity time as the temporal database holds the time variant

International Journal of Innovative Technology & Adaptive Management (IJITAM)
ISSN: 2347-3622, Volume-1, Issue-6 , March, 2014

information. While typing question are simple sentence for the system Spell Check will be evaluated by which the wrong spell will be corrected automatically. The input used for this system can use both British as well as US English. This system is designed to support both. This system produces the same result set as the SQL Interface. Data dictionary is developed in which all possible words which are related to the system are maintained. It should be updated whenever new information added in the system.

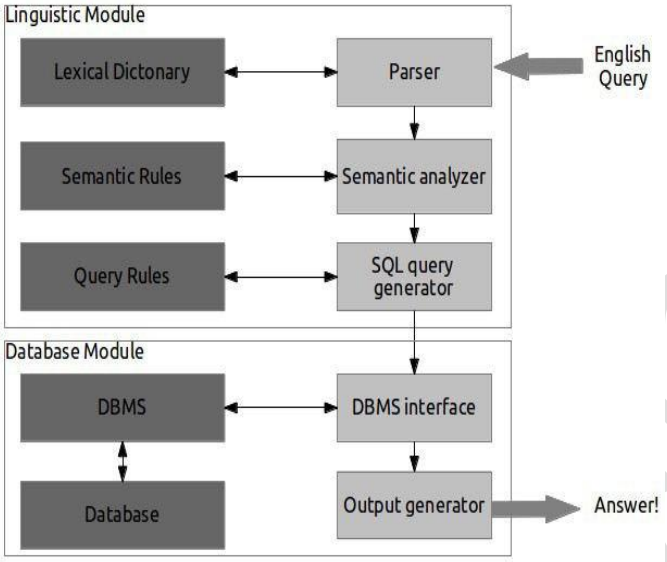

Fig 1. Architctural block diagram of the system

7.0 Evaluations

As case study for the feasibility of the NLP we proposed to implement the system in temporal database. In this comparison we compared with two different approaches, the first one NLP in temporal database using pattern matching produce the result which support for temporal data but it can retrieve results from single table only. The second is our approach we used the system for temporal data using probabilistic context free grammar by which it can accesses more than one table as well the temporal data. It also support for the simple complex queries. The performance impact of probabilistic context free grammar can be measured in three key areas:

a) Table Access
b) Complex Queries.

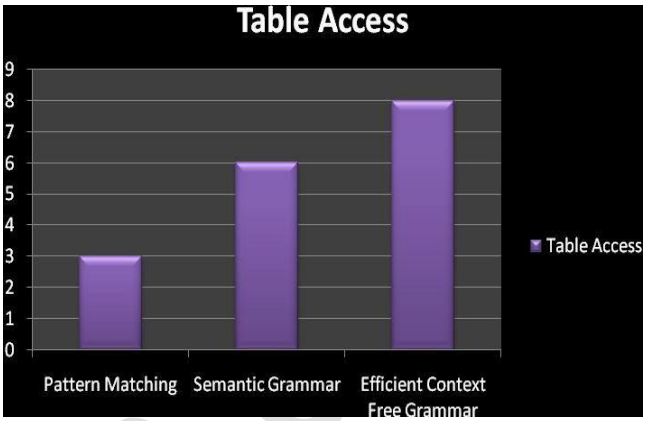

Fig 2. Performance comparison for table access

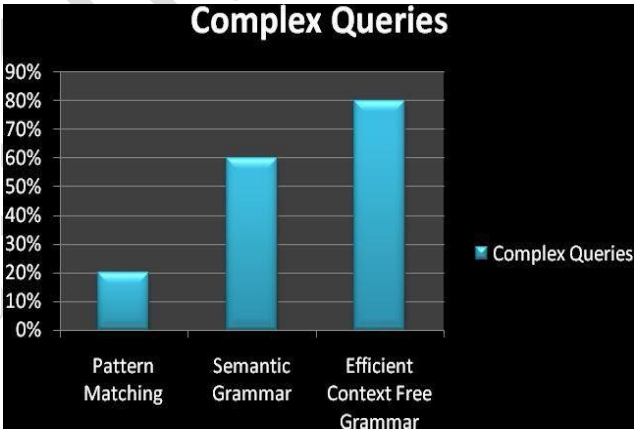

Fig 3. Performance Comparison for Complex Queries

8.0 Conclusion

In this paper human language query processing for temporal database has been designed and implemented to access temporal database. This lets the novice user to formulate their queries in their native language. The system can be focused for a specific domain, our can also be used as a generalized system. In this system we used temporal Database, as it is a time varying database we can formulate the historical data and also the data validity.

International Journal of Innovative Technology & Adaptive Management (IJITAM)
ISSN: 2347-3622, Volume-1, Issue-6 , March, 2014

9.0 References

1. *Abraham T and Roddick JF (1999) Survey of spatiotemporal databases. GeoIn-formatica. 3(1), 61–99.*

2. *Androutsopoulos I, Ritchie G and Thanisch P Masque/sql (1993) A client and portable natural language query interface for relational databases. Database technical paper, Department of AI, University of Edinburgh.*

3. *Gauri Rao, ChanchelAgarwal, SnehalChaudry, NikithaKulkarni and Patel SH (2010) Natural language query processing using semantic grammar. Int. J. Comput. Sci. Eng . Vol 02 No 02 219-223.*

4. *Gauri Rao and Patel SH (2009) Natural language query processing. Int . J. Comput. Appl. Eng. & Technol. & Sci. Vol 6 No. 2 495-505. 5. HuangiGuiaogZangi and PhilipC-Y Sheu (2008) A natural language database Interface based on probabilistic context free grammar. IEEE Intl.Workshop on Sematic Comput. & Sys. 155-162.*

6. *Jaymin Patel (2003) Department of computing, Imperial college, University of London M. Eng. Temporal database Sys. Individual Project on 18th June.*

7. *Piero Andrea Bonatti Elisa Bertino and Elena Ferrari Trbac (2001) A temporal role-based access control model. ACM Trans. Information & Sys. Security . 4(3), 191–223.*

8. *Ramasubramanian P and Kannan A (2004) Temporal event matching approach based natural query processing in temporal databases. Int. J. Information Technol.10(1), 88-100.*

9. *Tansel, Cliord, ShashiGadia, and Richard Snodgrass (1993) Temporal databases: Theory, Design and Implementation. Database Sys. & Appli. Series . Benjamin/Cummings, Redwood City, CA, 2nd ed. 633- 640. 10. Tsz Cheng S and Gadia SK (2002) Member IEEE Computer Society The Event Matching Language for Querying Temporal Data. IEEE Trans. Knowledge & Data Engg. 14(5), 1119–1125.*

11. *VijayalakshmiAtluri and Avigdor Gal (2002) An authorization model for temporal and derived data: Securing information portals. ACM Trans. Information & Sys. Security. 5(1), 62–94.*

12. *Winiwarter W and Ismail Khalil Ibrahim (2000) A multilingual natural language*

interface for ecommerce applications. Ph.D. thesis, University of Vienna, Austria. ACM 26(11):832-843