Indian Journal of Science and Technology Vol. 5 No. 6 (June 2012) ISSN: 0974- 6846

Intelligent query processing in temporal database using efficient context free grammar

K. Murugan[1] and T. Ravichandran[2]

[1]*Karpagam University, Coimbatore, Tamil Nadu-641021, India.*
[2] *Hindusthan Institute of Technology, Coimbatore, Tamil Nadu-641032 India.*
dhuvash_2004@rediffmail.com, dr.t.ravichandran@gmail.com

Abstract

Now a day's interaction with computer become essential, efficient processing, storing and retrieving of data from database will play very important role in the database application. To access the database the user should have a strong knowledge in SQL commands and procedures. The conventional database will give only current data but not past or future data. In this paper we propose an Intelligent Query Processing (IQP) for temporal databases. This will facilitate the novice user to interact temporal database in their native language (English), without using any SQL command or procedures. The main purpose of IQP is for an English sentence to be interpreted by the computer and appropriate action taken. Asking questions to databases in natural language is a very convenient and easy method of data access. The temporal data will support for past, present and future data. In temporal data we used third axis as time interval, which support both transaction time as well valid time. The valid time is the actual or real world time at which the data is valid. This paper proposes the architecture for translating English Query into SQL using efficient context free grammar. This system has been implemented using Java that can be used in any operating system and has been tested with data from the industry domain.

Keywords*:* Intelligent query processing, Temporal Database Context free Grammar .

Introduction

The Intelligent Query processing will play vital role in computer interaction. It is a part of Artificial Intelligence (AI) which has information retrieval, Machine translation and Analysis (Gauri Rao, 2010). The main aim of the IQP is to facilitate the novice user to interact database by avoiding the complex command and functions. This IQP will make the people easy to learn and use the computer as well (Gauri Rao & Patel, 2009). This will make the user to enter the text message as they would pass to the person. The interactive with computer is very essential and also more effective. Nowadays, computerization is implemented in almost all the fields. Particularly in medical field if the doctor wants to interact with database, he should know the complex command as well as procedure. But this IQP made everyone to access the database easily.

The conventional database systems are responsible for the storage and processing of huge amounts of information. The data stored in these database systems refers to information valid at present time. The conventional database does not provide models to support and process the past and future data. The temporal database stores data relating to time instances. It offers temporal data types and stores information related to past, present and future time. In temporal database the time period is added to express when it should be valid and when it is stored.

A database that can store and retrieve temporal data, that is, data that depends on time in some way, is termed as a temporal Database (Huangi Guiaog Zangi & Philip Sheu, 2008). The conventional database is generally two dimensional, and contains only current data. The two dimensions are rows and columns that interact with each other at cells containing particular value whereas temporal databases are three-dimensional with time interval as the third dimension. Temporal databases can also be referred to as time-oriented databases, time varying databases, or historical databases (Ramasubramanian & Kannan, 2004). A true temporal database is a bi-temporal database that supports both valid time and transaction time (Jaymin Patel, 2003). Transaction time is the actual time recorded in the database at which the data is entered and the time is known as the Time-stamp. Time-stamps can include either only the date or both the date and clock time. Time-stamps cannot be changed.

The other major type in temporal database is the valid time. Valid time is the actual or real world time at which point the data is valid (Tsz Cheng & Shashi Gadia 2002; Vijayalakshmi Atluri & Avigdor Gal, 2002). Conventional databases represent the state of an enterprise at a single moment of time. The conventional database holds the snapshot data. There is a growing interest in applying database methods for version control and design management in e-commerce applications, requiring capabilities to store and process time dependent data (Winiwarter & Ismail Khalil Ibrahiml, 2000; Piero Andrea Bonatti Elisa Bertino & Elena Ferraril, 2001,). Moreover, many applications such as medical diagnosis system, Forest Information Systems, Weather Monitoring Systems and Population Statistics Systems have been forced to manage temporal information in an ad hoc manner and support the storage and querying of information that varies over time temporal database holds time varying information, required by the above-mentioned applications. In the present scenario, writing better database queries for databases pertaining to an

©Indian Society for Education and Environment (iSee)

Indian Journal of Science and Technology Vol. 5 No. 6 (June 2012) ISSN: 0974- 6846

organization involves a significant amount of time and expertise. It has become a research issue now to increase the service capability of the database systems to help novice users to formulate a query for database access.

High-level query languages such as SQL are available in commercial databases. These are easy for those users with thorough understanding of programming concepts, database schema and relational algebra (Abraham & Roddick, 1999). To help non-expert users to perform query, a natural language front end is required. For those users who feel SQL difficult to use and for novice users who would like to retrieve data without having to learn querying mechanism such as SQL, a temporal natural language querying mechanism has been provided to access data from temporal databases (Tansel *et al.,*1993). The natural language Interface helps the distribution of the thought process from the human query users to the system (Androutsopoulos & Ritchie, 1993). Doing so helps reducing the effort spent by the query users in forming the queries.

Related work

Natural Language Processing (NLP) for the database is the best research work in NLP since it began. Interacting with the database in human language is very convenient and simplest way of accessing data, particularly for the novice user (who does not know about SQL commands and procedures). As NLP works well in single database it will be applicable for any of the following applications. Medical Diagnosis System, Forest Information Systems, Weather Monitoring Systems and Population Statistics Systems.

NLDBI this system used Efficient Context Free Grammarto parse the given queries. This system support to handle simple queries with standard joins condition (Gauri Rao, 2010). LUNAR involved system which answered question about rock samples brought from moon. In this Augmented Transition Network (ATN) Parser and Wood's Procedural Semantics is described. The system was demonstrated at the second annual Lunar Science Conference (Gauri Rao & Patel, 2009).

LIFER/LADDER was one of the best NLP database systems. In this system Natural Language Interface was implemented for database of information about US Navy Ships (Huangi Guiaog Zangi & PhilipC-Y Sheu, 2008). TEML design System acts as a standalone language for querying the temporal data. It can also be integrated with SQL for querying the temporal database (Ramasubramanian & Kannan, 2004). Design and Retrieving of data from temporal database was presented by (Jaymin Patel, 2003). Object evolution in a temporal database can be querying using an even matching language presented by TszS. Cheng and ShashiK. Gadia (Tsz Cheng & Gadia, 2002).

System descriptions

Brief information about the system is as follows:

Let us consider any databases, for example MicrosoftSQL in this we created nearly five Tables. All tables are normalized. Any users who may want to access data from these five tables are supposed to be familiar with SQL commands, procedures to generate query for getting the result set from the database. But my system support novice user to interact with database for accessing the data in their human Language. Consider the example to list the medicine from the table information for a particular disease, then we should create Query i.e. select medicine from information where disease = 'Cold'. But the user (he/she) who doesn't know MY SQL may not be able to access the database. Thus user should know the SQL commands and procedure. But it is not possible for all types of users. To avoid this difficulty we can use this IQP. All types of users can access data in the database. In the above said query is given in normal text in English language. For example "List the medicine for cold". Here both intelligent query processing statement and SQL Statement produce the same result. In this system novice user can easily access the databases.

Proposed system

Here the IQP 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 information. While typing question are simple sentence for IQP Spell Check will be evaluated by which the wrong spell will be corrected automatically. The input used for this IQP 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

Architecture

The architecture of the IQP and access control system is depicted in the Fig.1. It is briefly explained below.

Intelligent query processing

It enables the users to interact the database in their native language. It accepts the question in English and forward to the SQL query translation, then SQL statement interact the database and produce the result set.

Query Interface

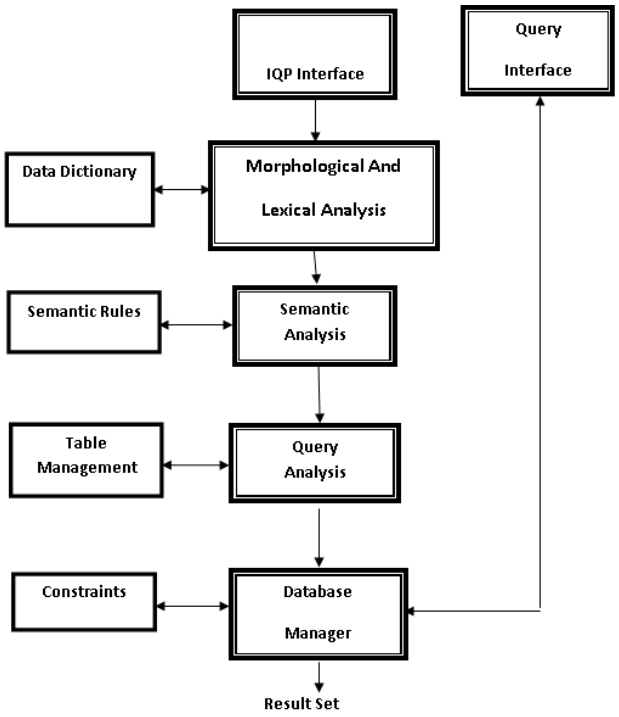
In this interface, directly gets the SQL query and then interacts with the database and produce the result set.

Morphological and lexical analyzer

The given input text is broken into several tokens and assigned the respective types with help of dictionary. Every token will be assigned their type such as noun, verb etc., with help of data dictionary.

Indian Journal of Science and Technology Vol. 5 No. 6 (June 2012) ISSN: 0974- 6846

Fig.1. IQP architecture



Semantic analyzer

The semantic analyzer checks the lexically correct input for grammatical correctness. The keywords are extracted and passed on to either RTL or the query analyzer. The output of the semantic analyzer will be whether the given input is syntactically or grammatically correct as per the given rules. English grammar rules are used for checking the syntax. The semantic analyzer has been formed based on the grammar for English language. Parser has been implemented specially for the use of this human language interface and uses SLR algorithm.

RTL analyzer

The next phase is the Relational Token List (RTL) analyzer, in which it generates the list of related tokens with help of data dictionary. It assigns a rank value for the related token.

Query analyzer

It takes input as the token list from RTL analyzer. In this process the actual meaning of the sentence should be identified. According to the ranked list, the transformation of domain specific is performed. The relational operators are replaced by the particular operators. As the last steps the classifications are used for framing appropriate SQL query.

Data Collection

This query will interact with database and produce the result set as output.

Algorithm

(a) Scan the given Input string from left to right, (b)While scanning spell check will be processed simultaneously, (c)Break the sentence into several tokens by eliminating the delimiters by which the meanings of symbols will be defined by the grammar, (d) Make interrelated token list, (e)Assign rank value for the tokens, (f)Create parsed syntax tree, (g)convert the leaves of tree to particular SQL,(h) Dependency structure used to accumulate information,(i) Split the query and extract patterns, (j) Look for sentence connectors,(k). Split query on the basis of connectors, (l) Specify the conditions in query by using the criteria, (m) Find attribute and values, (n) Check for validity of the data, (o) Translate into SQL Query,(p)Map values with respective tables.

Efficient context free grammar

Efficient context free grammar (ECFG) inherits most Advantages of CFG, and it is the easiest arithmetical representation to examine natural language. Usually natural language sentence are changed into a tree structure through ECFG, and the grammar tree is analyzed according to user's necessities. A efficient context free grammar consists of the following:

A terminal set: {wk}, where wk is a word, equivalent to a leaf in the grammar tree; A non-terminal set: {Ni}, Ni, which is assign used to produce terminals, equivalent to a non-leaf node in the grammar tree; Consider a sentence w1m that is a sequence of wordsw1 w2 w3……wm (ignoring punctuations), and each string win the sequence stands for a word in the sentence. The grammar tree of w1m can be generated by a set of pre-defined grammar rules. As wi may be generated by different non-terminals, and this situation also appears when generating anon-terminal (a non-terminal may be generated by different sets of several non-terminals), usually more than one grammar tree may be generated. Take sentence "ate dinner on the table with a fork" for example, there are two grammar trees corresponding to the sentence. It makes sense to select the most probabilistic grammar tree to proceed. However it is impossible to list all possible grammar trees then compute their probabilities and select the most probabilistic one, because there will be exponentially many of such trees.

Evaluations

As case study for the feasibility of the NLP we proposed to implement the IQP 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 IQP 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.

©Indian Society for Education and Environment (iSee)

Indian Journal of Science and Technology Vol. 5 No. 6 (June 2012) ISSN: 0974- 6846

Fig. 2. Performance comparison for table access

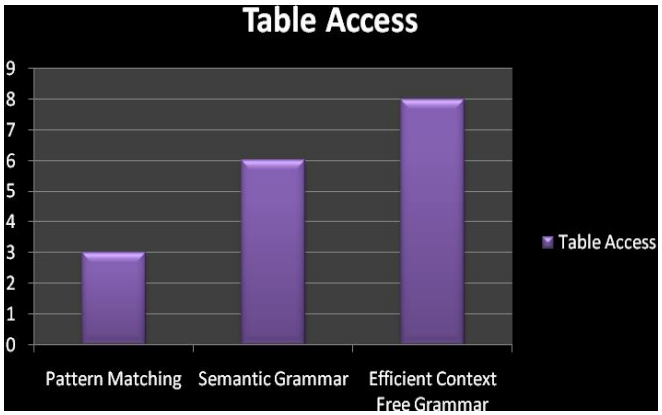
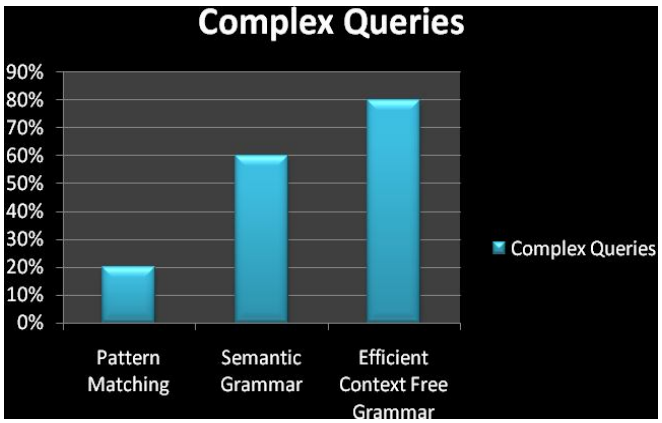


Fig. 3. Performance Comparison for Complex Queries



Flowcharts

The different types of operations which are performed during the human language process are shown in the Fig.2. The inputs to the flowchart have both the queries i.e. Normal query and also the human language query. When the input enters, the spell checker will be processed simultaneously. After getting the input it undergoes check for IQP, if it is false then it is normal query, so it will be passed directly to query process. If it is true then the IQP is undergone the process of lexical, syntax and semantic analysis. The error handling is done accordingly for all process. After the analysis process it will send to query process, there the query will be processed and then the results will be displayed. A schematic representation of the flowchart is shown in Fig.4.

Process description

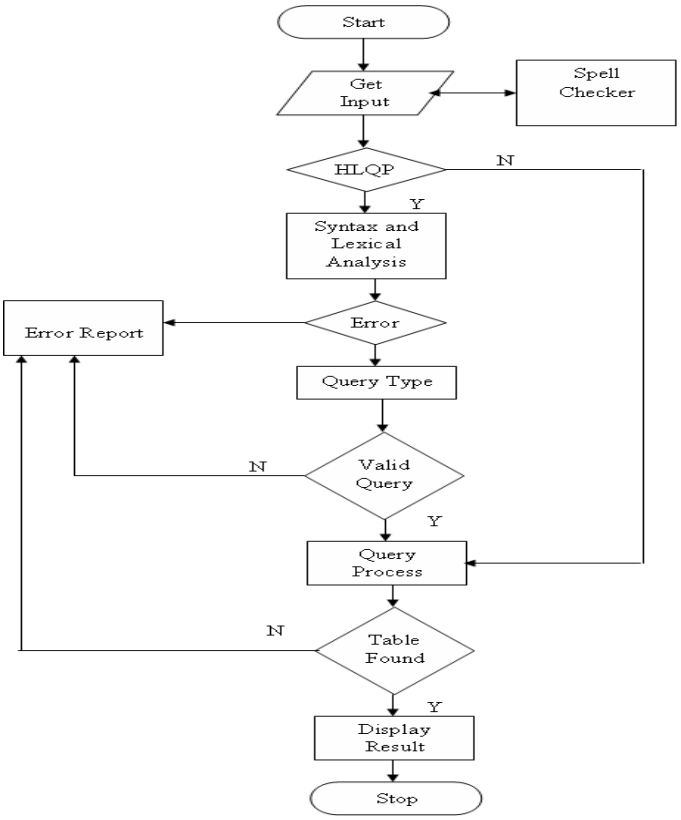
The given English Sentence is given into the Lexical, Syntax and Semantic analysis, by which the sentence is split into token. Then it will examine for grammar rules to frame ranked list. Then it is passed for query generation there the attribute, tables names will be identified and checked for the conditions and joins. Finally SQL query will be generated and passed to temporal database and produce the results.

The query interaction with temporal database should have the following process: First it should check whether the table is found or not, then have to undergo constraints check with help of constraint manager and will retrieve the results from the Database.

Let us consider the question: “Show the symptoms and medicines for cold”, then it will undergo all the analysis one by one and finally comes for probabilistic context free grammar and returns the following representation

(Select symptoms tablet (from Information (where disease=cold)))

Fig. 4. Flowchart



Now it passed to the query processor for query generation, there the phrases of the grammar is identified by the respective attributes and table. Then appropriate replacement will be carried out, by which the final SQL query will be framed.

The main objective of this IQP with temporal database is to enable novice user to interact the database easily. For this the system is designed with a dictionary which holds all the related words and also the table names. This facilitates the user to make the question in their human language in various forms.

As we used temporal database, we ensure the validity as well as the transaction time, by which we can retrieve the history of data or the transactions. In this temporal database we introduced third dimension as time axis.

Indian Journal of Science and Technology Vol. 5 No. 6 (June 2012) ISSN: 0974- 6846

Fig. 5. IQP Interface

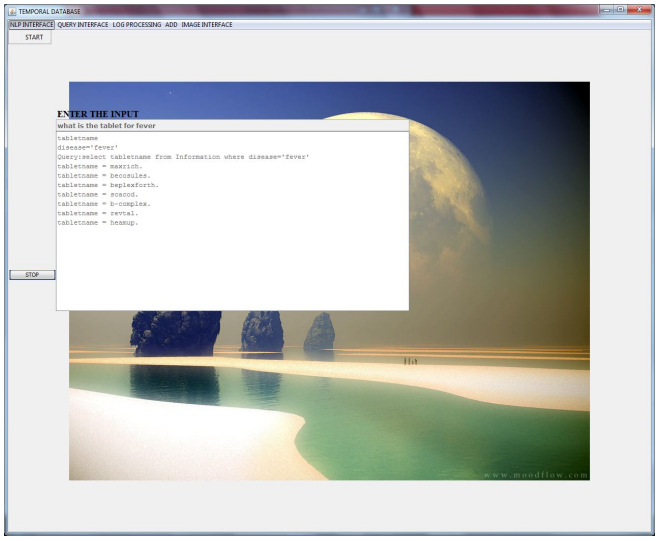


Fig. 6. IQP Image Interface form

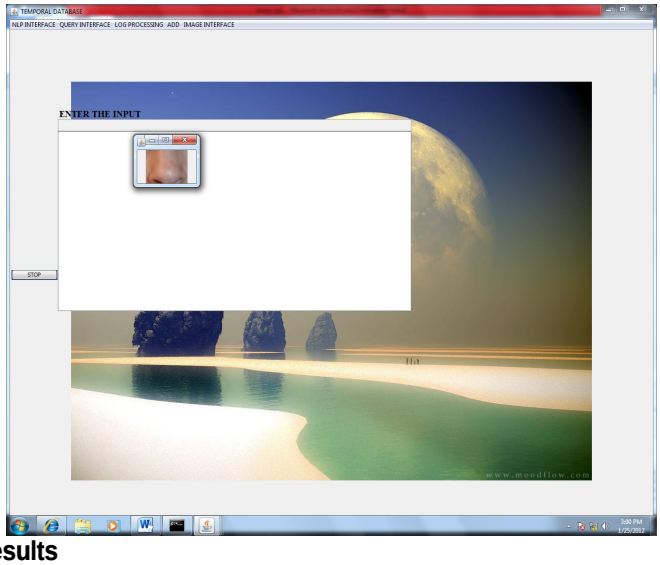


Fig. 7. IQP Entry form

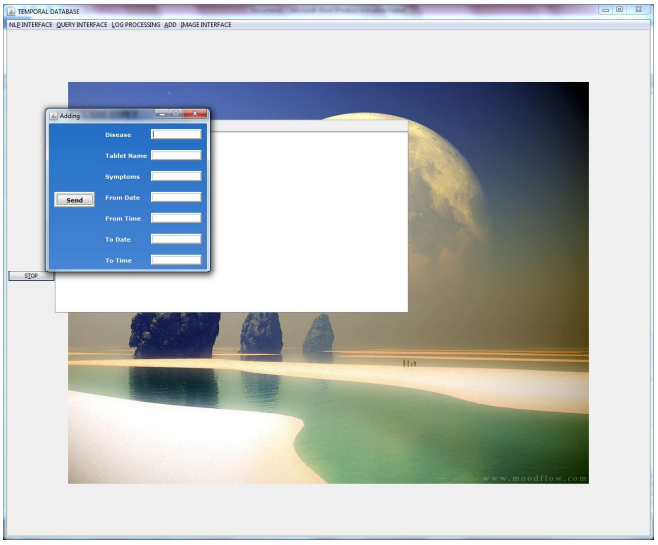
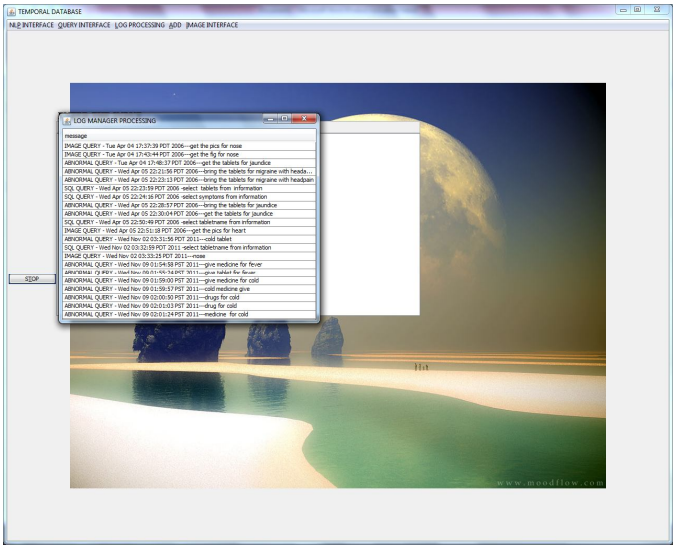


Fig.8. IQP Log details



Results

The graphical user interface is designed which is shown in Fig.5. The user should login by entering the username and password. The process starts now and the database connection will be processed. Now we should choose the GUI interface for IQP, in which the user enters the question in English and press the enter key. The SQL query will be generated and processed and it fetches the results from the temporal database and displayed in the text area. Similarly to process the normal SQL query we should enter the query in the appropriate interface which interacts with database and produce the results as shown in Fig.5, Fig. 6, Fig. 7 & Fig. 8.

English query

List the medicine for the fever

SQL query

Select tablet name from information where disease = 'Fever'

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 main purpose of this system is focused for medical domain, but this is a generalized system i.e. it also supports population system, Accounting System, Banking System, etc. 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.

Future directions

Increasing the size of the dictionary and grammar rules would increase the efficiency. Global dictionary will be introduced for various domains. Further research in this will enhance for the complex queries and all types of Joins.

Indian Journal of Science and Technology Vol. 5 No. 6 (June 2012) ISSN: 0974- 6846

References

1. Abraham T and Roddick JF (1999) Survey of spatio-temporal 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

©Indian Society for Education and Environment (iSee)