Rule based Domain Specific Semantic Analysis for Natural Language Interface for Database

Probin Anand
M Tech Scholar
All Saints' College of Technology, Bhopal

Zuber Farooqui
Asst Professor
All Saints' College of Technology, Bhopal

ABSTRACT
A database is defined as collection of information that is organized to access, manage, and update data easily and efficiently. All our data is stored in a database and there are multiple ways to interact with the database to access our data. A user needs some technical knowledge to extract data from the database. They need to use SQL for data definition, data manipulation, or data control. However, most of the users who need to extract data from a database are not technical experts. Therefore, there is a huge communication gap between the database and its core user. With the evolution of NLP a user can now talk to their database in their natural language without having to learn the language of the database. The communication gap between the user and the database has started to vanish with this amazing capability. In this paper, I will show you how to develop an effective and simple interface for a non-technical user to interact with their database in their natural language. I have chosen English as the user's natural language as it's the most commonly used language in the world.

Keywords
Natural Language Processing, Natural l anguage Database Interface, OPEN NLP, CRF

1. INTRODUCTION
Databases are found everywhere in the current information technology applications. For example, they are used to store structured data and information related to College related application, Restaurant related applications, Ticket booking related applications, Search Engines, etc. All these database access are limited from the normal end user in the form of Menu or form based user interfaces which may not be effective in most of these applications for complete usage user need to have an understanding of Structured Query Language (SQL) [8]. A user friendly interface is necessary to access these databases. This is because; Real time applications has large number of tables and columns(fields) and to the end user we can only give a menu based interface that shows the results from the database, we cannot fulfill all the user requirements in the menu based interface. Recently, usage of mobile phones has been increased significantly. Due to the compactness of the interface we need alternative for the menu based interface so as user can get the desired output from the database without having much knowledge of the database. Using a Natural Language Interface, we can query the database easily by issuing a command using speech or text. Also, in software industries, there are two main phases of building software systems [3]:

- Backend Development + Testing Phase (Design Database as per requirement and write SQL queries as per requirement)
- Frontend Development + Testing Phase. (Design the user interface and display the SQL queries output on to the screen or vice versa)

In general, backend development and frontend development teams are different. The database table names and the complex queries involving multiple joins written by the backend development team are difficult to understand for the frontend development team. The frontend development teams unaware of the logic used in SQL queries work on embedding them into a web application to create the desired interface. Both development and testing at backend is done by same team whereas testing at the frontend requires some basic understanding of backend SQL queries. For example, if there is any error in the frontend integration, the frontend team needs to identify whether the error is due to:

- Incorrect SQL queries, or
- No data in database tables in fact having correct SQL queries or
- Incorrect Frontend application

2. RELATED WORK
Databases usually deal with bounded domains and a rule based natural language can solve the ambiguity problem that can be caused due to natural language processing successfully (Gauri, 2010). Below are few existing Database Natural language Processors:

Lunar Science Natural Language Information System (LSNLIS or LUNAR) [11] **(**Woods, 1973) is a question-answering system developed for the geologists who were studying about rocks on moon and was the first system based on the concept of NLIDB. Geologists having no programming skills can retrieve data about the rock samples brought back from the moon in a lesser time and without any extensive cost. It was used to obtain information using the Apollo Mission. It was a waste of time and cost to teach the geologists the programming skill to process and retrieve data. The chemical analyses and the literature references where used as database and augmented parser and Procedural Semantics where used to retrieve the data for the questions asked by the geologists.

- RENDEZVOUS System (1977) takes the user input in the form of paraphrasing and clarification dialog in case the system was not able to parse the input to improve the system.
- LIFER/LADDER is similar to LUNAR and described by Hendrix (1978) was designed to access the database of information's of US Navy ships as a natural language interface. It was one of the first good among then existing systems. It queries a distributed database by using a semantic grammar to parse questions and give the resultant output.
- CHAT-80 (1980) best known NLIDB of the early 80's was developed in Prolog language. In CHAT-80 user input is converted into prolog expressions, which were evaluated against the Prolog database? several other experimental NLIDB's have their base from the code of CHAT-80.
- ASK (1983) [5] gives an interactive interaction allowing users to teach the system new words and concepts at any point. ASK has its own built - in database and other computer applications making it a complete information management system ability to interact with multiple external databases and electronic mail program. Users interaction with the ASK system is via natural

language. The users request in English is generated to suitable requests in the underlying system by ASK transparently.

- NALIX (2005) is XML based natural language interface.
- PRECISE [2] has a compact and interesting interaction with the user in case when user has similar question with different parameter each time.
- First query: "what is the capital of India?"
- Second query "Sri Lanka?"

For the second time user is not required to ask the complete question as "What is the capital of Sri Lanka?" in place the user can only ask "Sri Lanka?" The result will be displayed to the user by the system interpreting and giving the desired result.

- English Wizard is a query tool for relational database based on natural language. It is one of the leading software products that translate ordinary English (natural language) requests into Structured Query Language (SQL), and then return the results to the client. English Wizard enables applications to understand everyday English requests for most database reporting tools and client/server and results in information, and also provides graphical user interface (Karande, and Patil, 2009).

3. SYSTEM DESCRIPTION

Human has developed computer and applications in it to make the work easier with the incorporation of Natural Language Processing human computer interaction can be enhanced further and get the user desired output [8] (Rao et al., 2010). Information has played an important role in our lives since a long time ago; most people before making a decision will try to get the information they need. Recently, with the growth of technologies such as laptops and computers, cellular phones, the Internet and personal digital assistants (PDAs), information can be accessed almost anywhere, at anytime, by anybody, including those people who not necessarily have a computer background. Among the data sources databases is one of the major sources of information. "Databases contain a collection of related data, stored in a systematic way to model a part of the world". Person needs to formulate a query in such way that the computer will understand and produce the desired output for extracting information from a database. However, those who lack a computer background are not able to write such queries, especially. The most common way to obtain information for people is by asking questions in their natural language. But, computers cannot understand this language without any help (interface), as it merely represents a sequence of meaningless characters. Nowadays, people attempt to bridge this gap by providing user forms; however, this solution is limited as by using user form we can't cover the wide diversity of questions that a user can ask. Another solution which is costly and has time constraints is giving the question to an expert. Yet all of these factors only increase the appeal of natural language interfaces to databases,

A common example of usage and implementation of NLIDB System can be seen as: consider a CollegeDB database which has been created with help of MSSQL for a college. The database used is property normalized to remove any anomalies. Our system enable that end user who are not familiar to SQL queries and can't write complex SQL query can fetch data using normal English language. Consider an example: Consider the example: "Who teaches NLP?" For Computer, the term "teaches" here is ambiguous since it can associate a subject to teacher as well as teacher to students. In contrast, a human will immediately know that "teaches" here refers to the teacher-subject relation because NLP is a "subject". But a person, who doesn't know TSQL database syntax, will not be able to access the CollegeDB database unless he/she knows the SQL. But using NLP, accessing the database will be much simpler.

Both the SQL statement and NLP (simply in English) statement to access the course table in the CollegeDB database would result in the same output.

4. SCOPE OF THE SYSTEM

The scope of the proposed system as follows,

- A "collegeDB" database (Fig1) is created in MSSQL for the pilot work which implements a Relational Database Management System (RDBMS)
- For accommodating wider group users input language is chosen to be English being the most commonly used language
- Predefine rules and Semantic frames needs to be created for mapping natural language to SQL query.
- A limited (pre defined) data dictionary (sample are given in Table1) containing rules for POS tag, database keywords and ambiguity related semantic frames a created for a particular system. The data dictionary can be updated to widen the scope and usage of the application.
- Split the question string in to tokens POS tags are found for this work and have used "openNLP" to indentify POS tag.
- Escape words are considered as those words that does not have any rule description in the defining database and this is used to remove excessive words from the user input statement.
- For performance the NLP rule are created and are stored in the format of a B+ tree [21] (fig 2).
- To identify all SQL keywords, table name, column name involved in predefined rules specific for each of the cases such as NER (Named Entity Relationship) Finder, database tables, defined relations, database column name and values or parameters are used. In this paper we have used a CRF[18] base approach to find the NER and database terms
- To develop an SQL Template. Semantic frames for each of the rules are defined and based on each of the rules and template and any predefined Pattern [20] the SQL query is generated.
- To construct an SQL query using SQL elements. With the use of Semantic frames, rules description, table and column mapping for database an algorithm has been developed.
- Semantic Frames[13] are defined and these semantic frames take care of the pre known ambiguity among the words while processing the natural language.
- Relational Ambiguity is taken care by Creating Semantic rule for each ambiguous relational entity.

5. SYSTEM ARCHITECTURE

The NLIDB system [1, 2, 5, 7, 8, 9, 12 and 17] generates SQL query to retrieve the data from the relational database by translating user's query entered in English Language. The SQL Output will then be displayed to the user. This NLIDB system specially develops for English language as an initial step as English being the most commonly used language across the world. Natural language question, entered in English, is converted into a SQL statement an algorithm has been developed efficiently for producing suitable answers. Technical perspective used for this approach is the programming being done in c# and with a backend database of MSSQL. The architecture of the NLIDB system is demonstrated in Fig3, which depicts the layout of the process included in converting user question in English into a syntactical SQL query to be fired on the RDBMS and getting answers from the database.

The Main System of a NLIDB system [3] Includes (fig3)

1. User Interface: This module takes input from a user either in text format or speech audio for NLP
2. Natural Language Understanding (NLU): Based on the rules and semantic frames this module understand the English language and identify key field for query generation
3. Database Query Generation: After NLU from the identified key fields this module converts the user input to database query language like SQL. From the SQL Query obtained data is retrieved from the database respectively.
4. Natural Language Generation (NLG) [19] The output Obtained from the database in a natural language and is displayed in this module. To the user , user interface module displays the natural language output.

The NLIDB system includes the following stages [3] (fig 4):

- User Interface: The System includes the following modules
- Graphical User interface: It allows the user to enter the question in a natural language.
- Auto help for the user input includes the user defined rules and alias names
- Syntactic Analysis extracts the linguistic information from the NL query:
- Word Check: It checks all the words in the user question against the data dictionary for its existence
- POS Tagging: Using OPEN NLP, it splits the question string into tags (tokens) and gives an order number to each token identified as defined in table 2 and 3.
- Excess Word Remover: It removes excessive words from the question string
- Semantic Analysis: identify the domain elements and the underlying relations for each phrase, connecting these elements with the help of semantic frames below are the steps
- Mapping rules: It maps the rules with user input question statement.
- Named Entity Finder: based on POS Tag the entity is identified and checked if exists in our database than mapping entity is also identified.
- SQL Elements identifier: It identifies all
- SQL elements involved in the user input statement by using rules and construct the SQL Template String (fig 6).
- Mapping SQL Template: It identifies correct SQL Template String algorithm for generating SQL query.
- SQL Query Generation: It constructs a query in SQL.
- Query Processing :
- Run the SQL Query: It gives the SQLquery generated to the back-end database.
- Data Collection: This module collects the output of the SQL statement and places it in the user interface screen as a result.

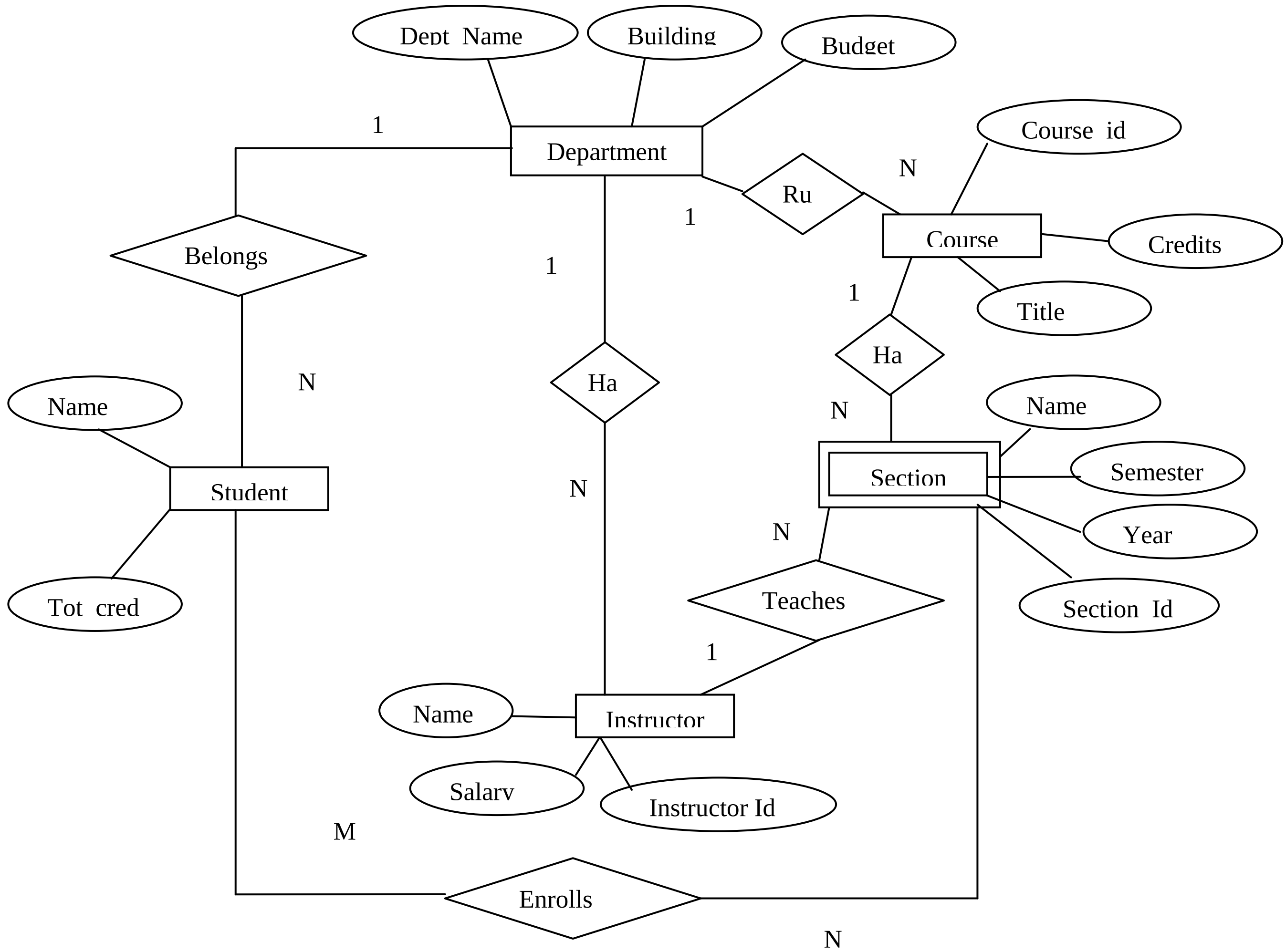


Fig 1: ER Diagram of a College Database

Table 1: Rule Definition in Tabular Form

Rule definition	Query Term(Rule Symbol)
select	Select
Choose	Select
Show	Select
Extract	Select

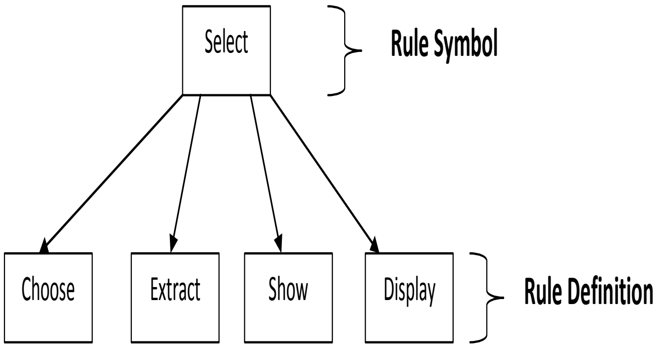


Fig 2: Graphical Representation of Rule Definition

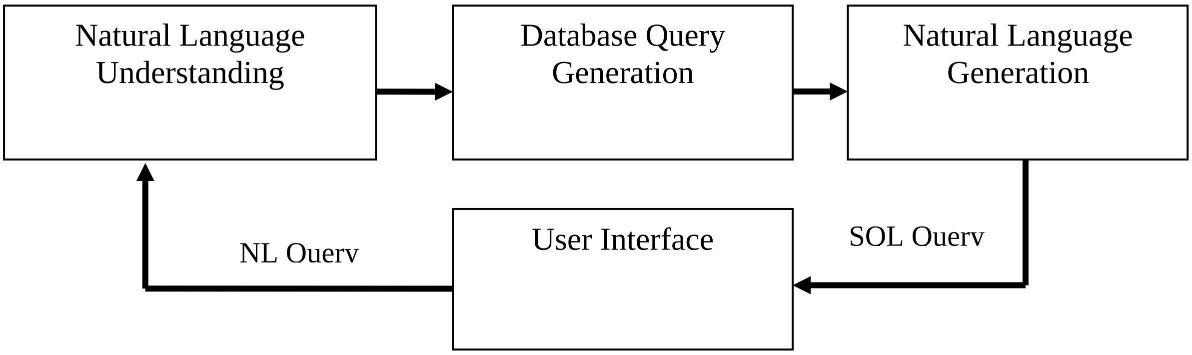


Fig 3: Components of a System

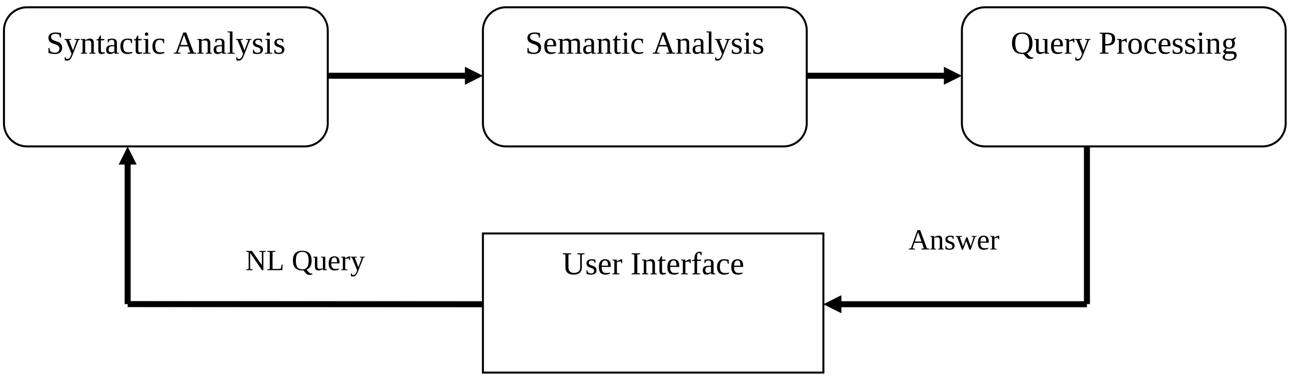


Fig 4: Stages in the system

Table 2: Chunk Abbreviation and meaning

Abbreviation	Detail
ADJP	Adjective Phrase
ADVP	Adverb Phrase
PP	Prepositional Phrase
PRT	Particle
UCP	Unlike Coordinated Phrase
CONJP	Conjunction Phrase
INTJ	Interjection
LST	List marker
SBAR	Clause introduced by a subordinating conjunction

Table 3: POS Tag Abbreviation and meaning

Abbreviation	Detail
NN	Noun, singular or mass
NNP	Proper noun, singular
NNPS	Proper noun, plural
NNS	Noun, plural
VB	Verb, base form
VBD	Verb, past tense
VBG	Verb, gerund/present participle
VBN	Verb, past participle
VBP	Verb, non-3rd ps. sing. present
VBZ	Verb, 3rd ps. sing. present
.	Sentence-final punctuation

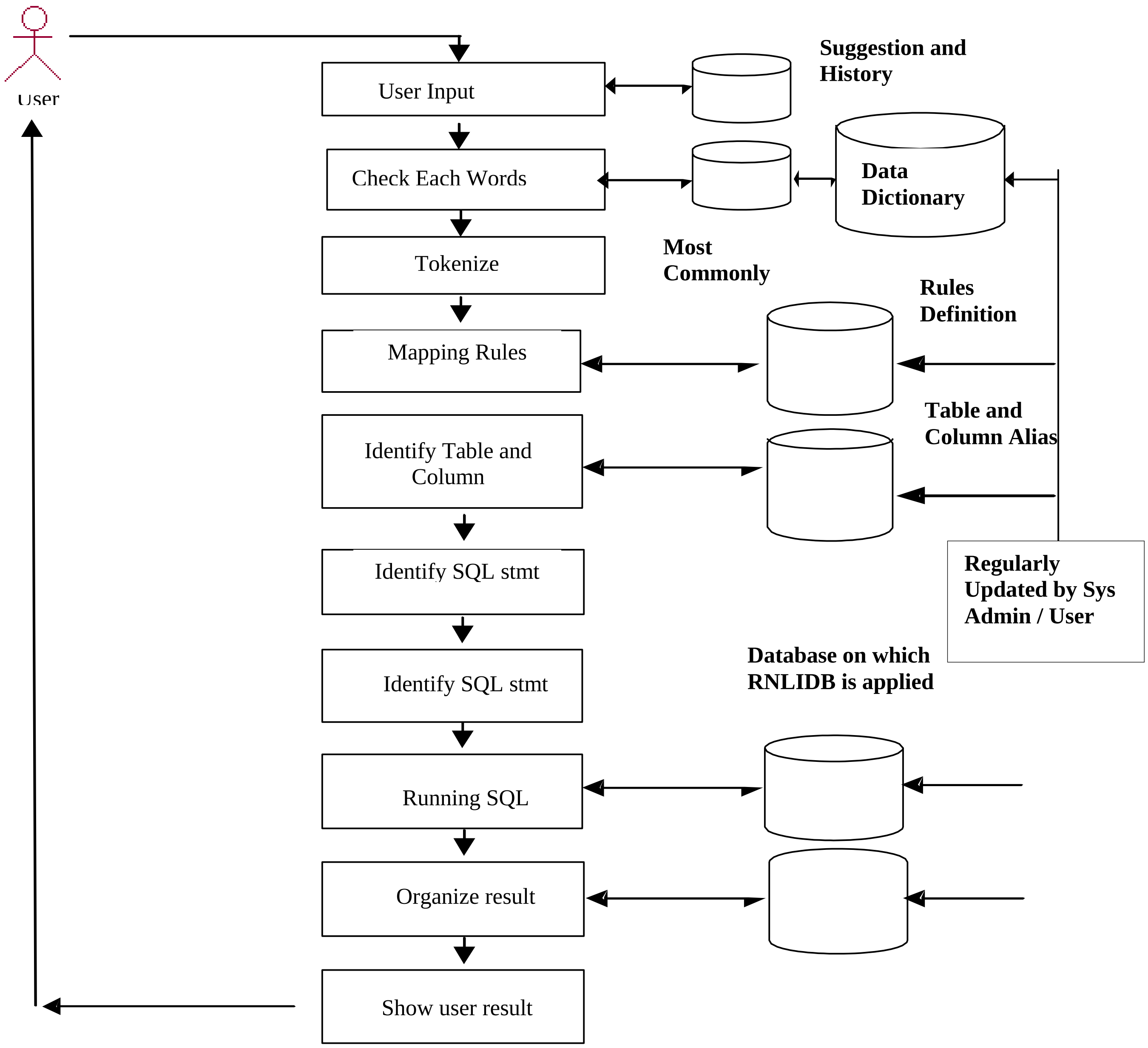


Fig 5: System Detail Architecture

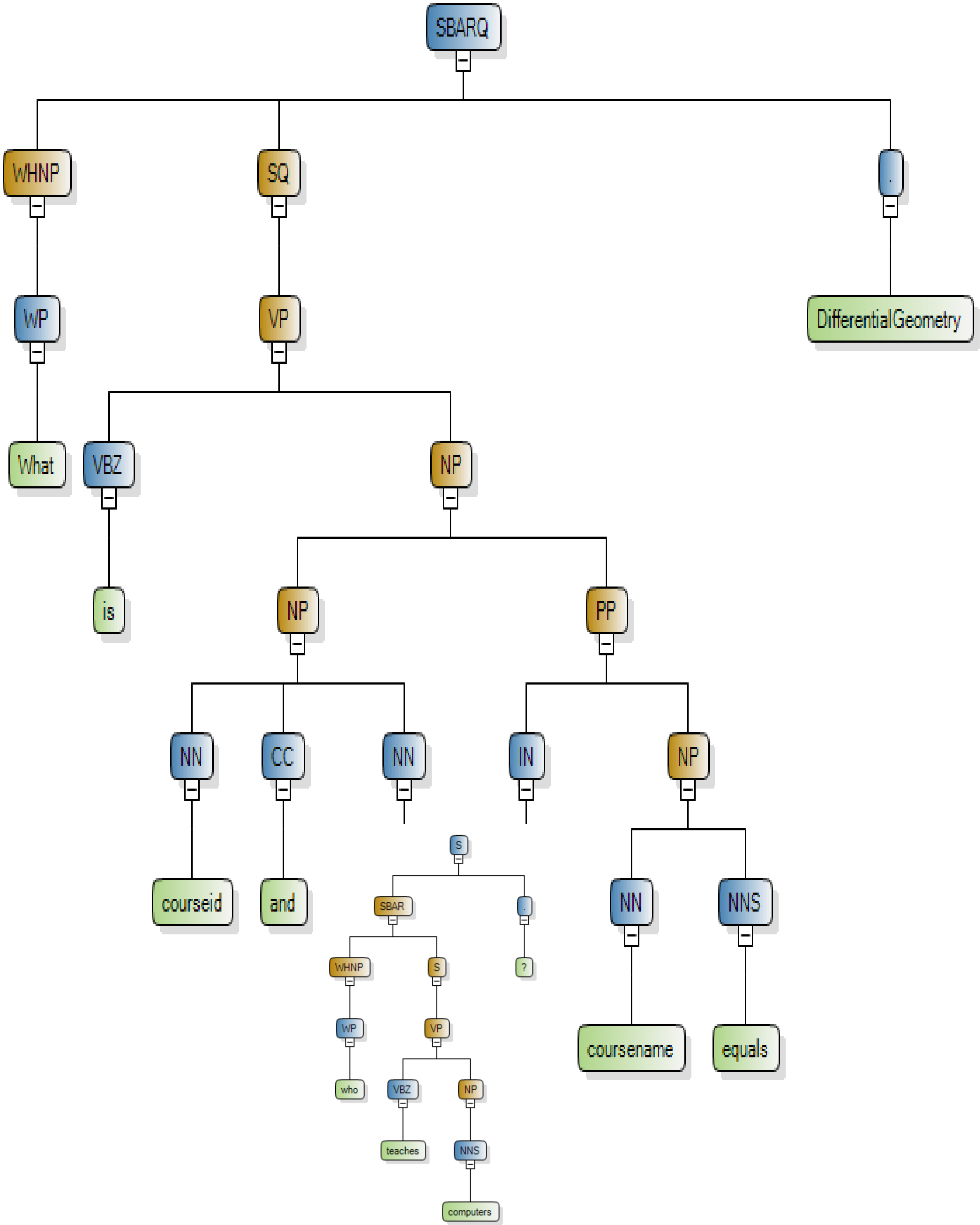


Fig 6: Parse tree generated for the NL string “Who teaches Computers?”

6. ALGORITHM APPROACH

The NL query (given by the user) from user's terminology to system's terminology. This procedure includes:

- Parse the User query
- Identifying the named entities using a NER tool.
- Check in the stored output in case of the content based system[22] for Entity and domain terms
- Correcting possible errors in named entities using mostly nearly matched.
- Finding synonyms to map lexical terms using CRF [18] to domain concepts.
- Find the Named entity
- Find the table Name
- Find the Column Name
- Find the Variable
- Generalizing the lexical terms to synchronize with domain terms.
- Based on the lexical terms generate the semantic Parse tree.
- Based on the based suited semantic argument for the domain lexical terms generate the corresponding SQL.
- Validate the SQL Script.
- Run the SQL Query to fetch the output.
- In case of Content based system store the output result for the next user interaction.

7. RESULT

In this paper result are obtained by carried out experiments on multiple Colleges related queries. This paper include single dialog and multiple dialog approach for our system. For the single dialog this paper have used NLP that involves complex join related queries and multiple join statements. And For the multiple dialogues this paper has used NLP to use wide range of topics such as course registration. Every multiple dialogues has a sequence of User-System interaction (or turns). On an average, each dialogue contains about 2 to 4 responses.

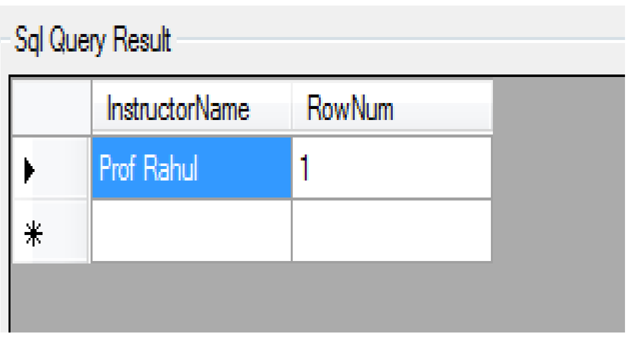


Fig 7: User Input NL Query

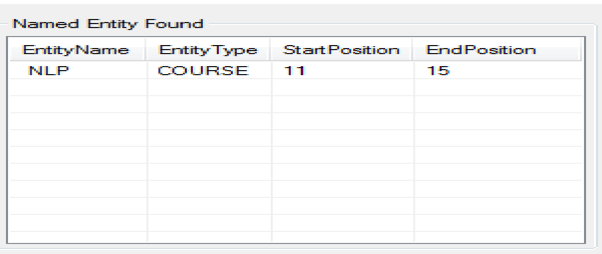


Fig 8: Found Entity in User Query

Out of User-System interaction, some are of strongly coherent type, some are of coherent type and few are of weakly coherent type. Like human to human interaction NLIDB user also try to interact with the system based on the previous interaction and in our system this paper have also taken care of such interaction this paper have set the flag and based on the nature of the flag, being true for the case of context based model and false for the rest of the case.

Context based model and the dialogues belong to contextual model. The above approach indicates that the method proposed in this paper is sufficient to identify contextual information in most of the real-time interactions handling this type of interaction along with the normal human system interaction.

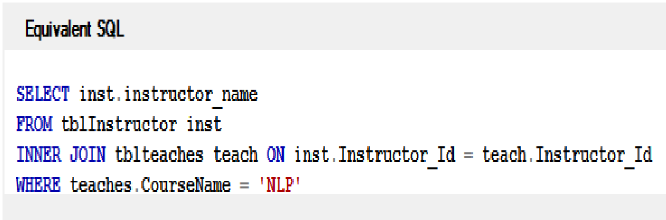


Fig 9: User Output

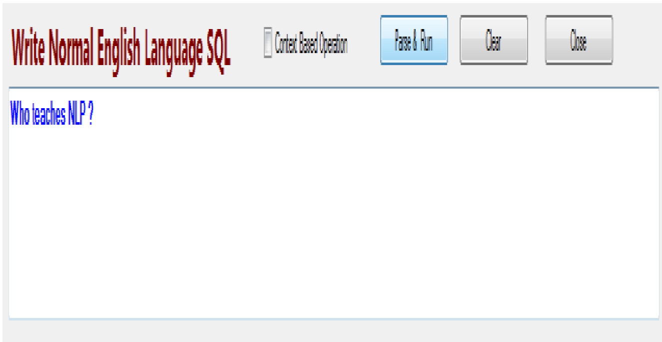
For a given NL query from the user this paper parse the string and obtain the below Output by finding the table name in the query string, the column and variables in the string. The Named entity is also found in the user query and based on all the found details a SQL query is generated by the software and the output is displayed accordingly. The ambiguity handled by the use of semantic frames:

Table 4: Semantic Frame for teach (take 1)

Role	Domain Term	Query Term	Value
Root	Teaches	Teaches	
S1	Instructor		Who
S2	Course		Computers

Table 5: Semantic Frame for teach (take 2)

Role	Domain Term	Query Term	Value
Root	Teaches	teaches	
S1	Instructor		who
S2	Student		Ram



In the example query "Who teaches Computer", the verb 'teaches' (take) can imply either teaching a particular subject or teaching a particular student. The "teach" term can have various meaning based

on the entity it is related to either subject or student. This ambiguity is resolved with the help of semantic frames using the predefined rules for the term "teach". For 'teach' the two possible frames as shown in Table 4 and Table 5. Based on the analysis our system is easily able to handle the ambiguity caused due the word "teach" and can create complex relation query for the similar NL query from the user.

8. CONCLUSION

Any Interface with the use of Natural Language Processing can be powerful enhancements, human language being so natural and considering English as in our paper which is most commonly used language across the globe. The Rule based domain specific semantic analysis Natural Language Interface for Database is no exception because through this system converts a wide range of text queries (English questions) into formal (SQL query) ones that can then be run against a database by employing generic and simpler processing techniques and methods. This paper have to define the relation involving the ambiguous term and domain specific rules and with this approach this paper can make a NLIDB system portable and generic for smaller as well as large number of applications. This paper also classify the human-system interactions and based on the types of interaction this paper proposed four models (Normal, Linear Disjoint, Linear Coincident and Non-Linear Model) based on the way of human interaction contextual information can be used where required in the interactions. Among the responses this paper proposed a relationship schema and flag and central approach this paper use these relationships to identify context based model and contextual information in the human system interaction. We evaluated our approach on College related queries and also define semantic analysis approach on few ambiguous term related to a college system. The results confirm the value of our system. In this paper, we have also handled the context based system along with the ambiguous term handling. This makes the method proposed in this paper more suitable and efficient one in identifying the contextual information in most of the real-time interactions. Defining semantic frames takes highest portion of time while porting a NLIDB system from one domain to another. In this work, we reduced this time to a great extent. However, Machine Learning approaches can be explored to automate the process of building semantic frames. A large amount of dialog corpus is needed for such automation

In this paper, only focuses is on context based interaction along with SELECT, FROM, WHERE and JOIN clauses of SQL query and also handling complex query that results from the ambiguous NL query. But for some NL queries, we may also need GROUP BY, HAVING BY, ORDER BY, etc clauses of SQL query a future research can be done for handling these types of query having complex join and also group by and order by clauses.

9. REFERENCES

[1] Mrs. Neelu Nihalani, Dr. Sanjay Silakari and Dr. Mahesh Motwani, (2011) "Natural Language Interface for Database: A Brief Review", IJCSI International Journal of Computer Science Issues, vol. 8, no. 2, pp. 600-608.

[2] R. Ahmad, M. A. Khan, and R. Ali (2009). "Efficient transformation of a natural language query to SQL for Urdu". In Proceedings of the Conference on Language & Technology, page p53.

[3] A. R. Akula, R. Sangal, and R. Mamidi (2013). "A novel approach towards incorporating context processing capabilities in NLIDB system". In Proceedings of the 6th International Joint Conference on Natural Language Processing (IJCNLP), pages 1216–1222.

[4] I. Androutsopoulos(1992). "Interfacing a natural language front-end to a relational database". Master's thesis, Cite seer.

[5] I. Androutsopoulos, G. Ritchie, and P. Thanisch (1993). "Masque/sql-a client and portable natural language query interface for relational databases". Database technical paper, Department of AI, University of Edinburgh.

[6] Katz, B., et al., (2002). Omnibase: Uniform access to heterogeneous data for question answering. In Proceedings of the 7th International Workshop on Applications of Natural Language to Information Systems (NLDB 2002).

[7] Minock, M., (2005). A Phrasal Approach to Natural Language Interfaces over Databases, proceedings of the International Conference on Applications of Natural Language to Information Systems (NLDB), pages 333-336.

[8] Rao. G., et al., 2010. Natural language query processing using semantic grammar, International Journal on Computer Science and Engineering (IJCSE),

Vol. 02, No. 02, 219-223

[9] Siasar djahantighi. F., et al., 2008. Using Natural Language Processing in Order to Create SQL Queries, Proceedings of the International Conference on Computer and Communication Engineering 2008, Kuala Lumpur, Malaysia

[10] Thompson, C.W., Pazandak, P., Tennant, H.R., .2005. Talk to your semantic web. IEEE Internet Computing 9(6), 75–78

[11] Woods, W. A.,. 1973. Progress in natural language understanding: An application to LUNAR geology. AFIPS Natl. Computer. Conj: Expo.. Conference Proc. 42, 441-450.

[12] I. Androutsopoulos, G. D. Ritchie, and P. Thanisch. 1995. "Natural language interfaces to databases-an introduction". arXiv preprint cmp-lg/9503016,

[13] Gauri. R., et al., 2010. Natural Language Query Processing Semantic Grammar. (IJCSE) International Journal on Computer Science and Engineering Vol. 02, No. 02, 2010, 219-223. http://www.enggjournals.com/ijcse/doc/IJCSE10-02-02-20.pdf

[14] Hendrix. G. G., et al.,. 1978. Developing a natural language interface to complex data, in ACM Transactions on database systems, 3(2), pp. 105- 147,

[15] Huangi, et al.,. 2008. A Natural Language database Interface based on probabilistic context free grammar. IEEE International workshop on Semantic Computing and Systems.

[16] Karande, N. D., and Patil, G. A.,. 2009, Natural Language Database Interface for Selection of Data Using Grammar and Parsing, World Academy of Science, Engineering and Technology.

[17] Rukshan Alexander, Rukshan Prashanthi, and Mahesan Sinnatham. 2013. "Natural Language Web Interface for Database". Proceedings of the Third International Symposium, SEUSL: 6-7, Oluvil, Sri Lanka.

[18] John Lafferty, Andrew McCallum, and Fernando Pereira. 2001. "Conditional random fields: Probabilistic models for segmenting and labeling sequence data". Proceedings of the Eighteenth International Conference on Machine Learning, pages 282–289

[19] H. Dalianis and E. Hovy 1996. "Aggregation in natural language generation". Springer.

[20] A.-M. Popescu, O. Etzioni, and H. Kautz.. 2003. "Towards a theory of natural language interfaces to databases". Proceedings of the 8th international conference on Intelligent user interfaces, pages 149–157. ACM,.

[21] X. Meng and S. Wang. Nchiql. 2001: The chinese natural language interface to databases. In Database and Expert Systems Applications, pages 145–154. Springer,.

[22] N. Bertomeu, H. Uszkoreit, A. Frank, H.-U. Krieger, and B. Jorg. 2006. "Contextual phenomena and thematic ¨ relations in database a dialogues: results from a wizard-of-oz experiment". Proceedings of the Interactive Question Answering Workshop at HLT-NAACL 2006, pages 1–8. Association for Computational Linguistics.