A Domain and Language Construct based Mapping to Convert Natural Language Query to SQL

Alka Malik
M.Tech (Scholar)
UIET MDU
Rohtak, Haryana, India

Rahul Rishi
HOD Computer Science Dept.
UIET MDU
Rohtak, Haryana, India

ABSTRACT
Database management systems (DBMS) have been widely used for storing and retrieving data. However, databases are often hard to use since their interface is quite rigid in cooperating with users. End user is required to issue SQL query to retrieve information from the database. Thus, a user cannot interact directly because of lack of knowledge about the SQL query form, thus is restricted to available application options. To provide the direct access of user on dataset, some unstructured query processor is required. The present work is an interface that will convert the unstructured text query to structured SQL query. It is an intelligent interface formation between the user and the dataset and defined in three main stages namely Descriptive Database Construction, Unstructured Query Filtration and Mapping Stage. The system provide the robustness in terms to handle the broader range of user queries and is be implemented in java environment on Enterprise Employee Database.

General Terms
Natural Language Processing, Natural Language Interface to Database.

Keywords
Natural Language Interface to Database (NLIDB), Keyword based Interface to Database (KBIDB), Structured Query Language (SQL).

1. INTRODUCTION
Database management system is the collection of interrelated data and set of programs to access and manages that data. To access the information from the database we need to have knowledge of Structured Query Language (SQL). But for computer illiterate it is not possible .Thus the need for using Natural Language as a means of accessing a Database arises. So is the need for Natural Language Processing (NLP). Natural Language Processing (NLP) is a field of computer science, artificial intelligence and linguistic concerned with the interaction between computer and human language [17].

NLP is a technique that makes a computer understands the languages that are native to humans. Thus understanding NLP there are multiple options of providing user interface to the Database i.e. Form based interface, Natural language based query interface, Keyword based query interface. Form based interface provides the interactive user interface but it gets failed if provider has not provided retrieval of data variations which is possibly manipulated by SQL. Natural language or Keyword based interface allow the user to access data by entering query in natural language either in English or in any other language. Natural language based query interface accept the query sentence and try to understand it by applying lexicon, syntactic and semantic analysis and then converts into SQL. Keyword based interface accepts query type of search engine query which retrieves keyword from the input query and converts into SQL by applying rules from the generated knowledge base[1].

Here is discussed another approach for NLIDB system, where domain and Language construct specific database is constructed to increase the query answering capability of the system. This database will contain the domain and Language specific constructs and their relation to represent all perspectives of user query. Hence the overall query processing capability is substantially increased as expected by the system. At first the query is filtered to extract the keywords, and then these keywords are identified as Domain or Language Constructs. Based on mapping database and identified constructs the SQL query is formulated and desired data is retrieved from the dataset.

2. LITERATURE REVIEW
2.1 Natural Language Interface to Database (NLIDB)
The most widely use of Natural Language Processing (NLP) is in the development of a natural language interface to database systems (NLIDNB). With the help of these systems the user can interact with database in a more convenient and flexible way. The aim of Natural Language Interface to Database is to provide an interface where user can interact more easily and efficiently using their natural language and access or retrieve information [3]. A person having no knowledge of Structured Query Language (SQL) may find himself or herself handicapped in corresponding with the database [4]. Natural language based query interface accept the query sentence and try to understand it by applying lexicon, syntactic and semantic analysis and then converts into SQL.

The components that are part of the Architecture of a NLIDB system can broadly be divided into the following:

2.1.1 Language or Linguistic Component
The language components perform the Morphological analysis. Lexical Analysis used for the identifications of properties of the words used in a sentence as noun, verbs, etc. WordNet is a useful tool for tracing connections between the words for morphological analysis and to be used as a lexicon, for performing lexical analysis [11], [12]. Syntactic and Semantic Analysis is used to convert the natural language sentences to parse tree for mapping to the query language. Stanford Parser [13] and Link Grammar Parser [14] can be used for the purpose. Also it performs Discourse Analysis and Pragmatics.

2.1.2 Intermediate Language Representation
ILR provides the interface between language components and database components. Intermediate language facilitate the

generation of SQL query making use of a knowledge Base [10]. DRS (Discourse Representation Structure) are an IL making use of Ontology based semantic interpreter to convert natural language queries to SQL queries.

2.1.3 Database Components

These make use of intermediate language and the domain knowledge bases available to facilitate the translation of the queries. 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. The knowledge about syntax is usually present in the linguistic component of the system, particularly 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 is 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].

2.2 History of NLIDB Systems

LUNAR a system informally introduced in 1971 that answers questions about samples of the rock brought back from the moon. For this it makes use of two databases using the Augmented Transition Network (ATN), for the chemical analysis and Wood's procedural semantics, for literature references [6].

GINLIDB [15] making use of semantic grammar support a wide range of natural language sentences. Using two types of semantic grammar each for distinctive purposes, one is single lexicon semantic grammar for lexicon non terminal words and another decides terminals that forms phrases or sentences using a composite grammar. GINLIDB also performs Syntactic analysis based on Augmented Transition Network (ATN), checking if the tokens' structure is in allowable grammatical structure or not.

CHAT-80 earliest known NLP system, implemented using Prolog. CHAT-80 processes the question into three stages such as a) It represents a word in logical constants, b) Representing verbs, nouns and adjectives with their prepositions as a predicate and c) Representing complex phrases or sentences by conjunctions of predicates on the database consists of facts about 150 of the countries world and small set of English language vocabulary [16].

Vietnamese NLIDB [18] is an interface developed to facilitate the survey of database for the data of economic survey by the individuals and businesses interested in knowing the information.

Hu Li and Yong Shi have proposed an approach to creating a WordNet-based natural language interface to relational databases which allows end-users to access and query information in database with a natural language. Here the approach integrates WordNet as the base lexicon and ontology as the knowledge base of the semantic interpreter. The presented framework is based on ontology techniques to implement portable NLIDBs that make it easier to migrate from one domain to another. It also defines relational database E-R model and domain business module by using OWL Ontology, which increases the accuracy of query sentences [16].

The system presented by Neelu Nihalani shows the mapping of natural language queries to SQL. She has proposed a general architecture for an intelligent database interface and also a real implementation of such a system which can be connected to any database. One of the main characteristics of this interface developed by the author is domain-independence. Another characteristic of this system is ease of configuration. The intelligent interface employs semantic matching technique to convert natural language query to SQL using dictionary and set of production rules. The dictionary consists of semantics sets for tables and columns [9].

The work of Vesper Owei defines a method of conceptual query filtration using natural language processing. The system work on natural language processing for the generation of structured query. It is an interface to process the query statement under the predicate analysis. Author defined the conceptual search using NL parsing for full fledged NL parsing [2].

A work on natural language processing on relation database defined by Alok Parlikar. Author defined the text to query processing under the SQL query analysis and processing. This work is the improvement work on XML processing for the structural analysis so that the database support will be done. It defined the work on question answering system where the questions are performed in text form and query mapping was proposed to derive the database results [8].

DBXplorer making use of two preprocessing steps called publish and search provides Keyword based search to the commercial relational database. Publish enables database for keyword search by building the symbol table and associated structures. Search gets matching rows from the published databases [19].

NUITS is a system that implements search algorithm using structure and content level clustering allowing the user to enter the simple, typed and conditional keyword based query and viewing the resulting tuples [20].

3. PROPOSED ARCHITECTURE

The proposed system is an approach to develop an NLIDB system. Here the robustness to the unstructured query processing is defined by using the improve natural language processing concept and the mapping of the language constructs to generate effective query. In this work, a three stage model is defined to perform the unstructured query processing on domain specific dataset. These three processing stages are defined here as:

Stage 1: Descriptive Database Construction

In this work, the organized data is presented in the form of relational dataset with some domain specific records. To perform the mapping with database, language construct database is constructed. This database will contain the following information

- Contains the domain specific information
- Contains the language construct under different categories
- Contains mapped words.

Stage 2: Unstructured Query Filtration
This stage is immediate to the user unstructured query input. This stage will filter the user query and extract the meaningful information such as keywords, query constructs etc. This filtration stage includes

- The pruning of the unstructured to remove the stop words
- The identification of the domain specific constructs
- The identification of language constructs

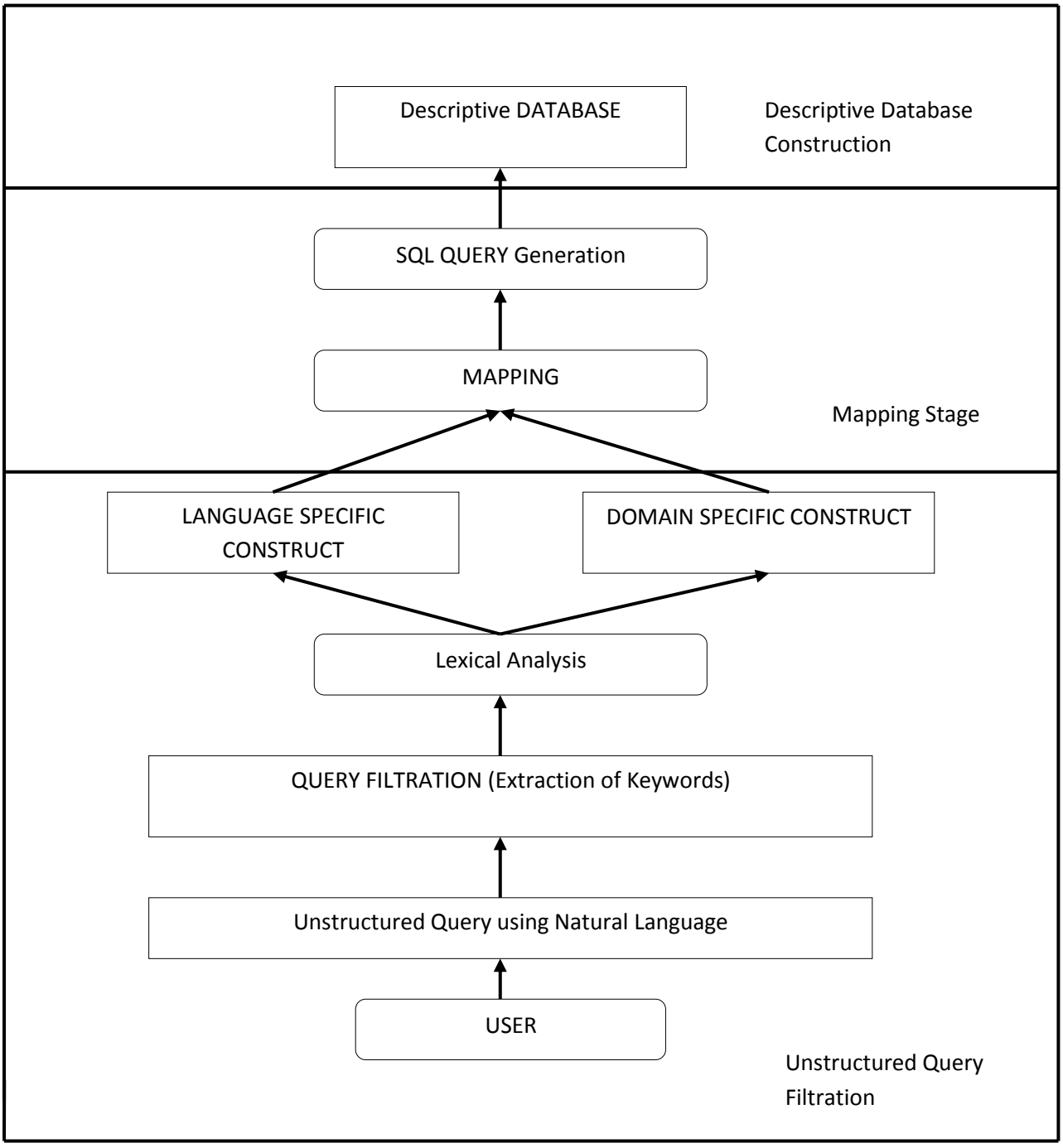

Fig 1: Architecture of a Domain and Language Construct based mapping to convert Natural Language to SQL.

Stage 3: Mapping Stage
This stage is the intermediate stage between the user query stage and the domain specific dataset stage. This stage will use the natural language processing to convert the unstructured query to structured query. The working of this stage include

- Convert the filtered unstructured query to SQL query
- Perform the query keyword with language constructs
- Convert the query construct to SQL constructs.

The "Figure 1" is showing the basic block diagram considered in this work as the main approach. The work will accept the user query in the form of unstructured text. The query is the form of some question performed over the dataset to extract

the information text. This input text will be filtered at the earlier stage; this filtration process will do two main tasks.

- First to remove the stop list words from the list and
- Later on identify the keywords under different categories from the document.

As keywords are derived, the analysis of these keywords will be performed under the language constructs and the domain constructs. This kind of mapping is performed based on the dataset keyword access. Once the domain and language constructs are separated, the next work is to organize all these keywords in an organized way. To arrange them, numbers of rules are defined under NPL. These rules will arrange the keywords in meaningful way. Once the organized or structured form of query is obtained, the next work is to obtain the SQL equivalent of this unstructured query. This query is processed over the dataset to generate the final result from the query.

The categorization scheme discussed here attempts to do general categorization, not based on, and completely independent of, a knowledge base. It categorizes questions based on the types of information in the question and the type of answer expected. This absolute separation from the knowledge base is logically important because a question asked in English should be categorized the same way no matter what knowledge the system being asked the question has. This is not to say that a question will not have different meanings and thus different answers based on the context of the question, but Author feel that this is merely a result of the way the knowledge base interprets the question. The answers returned could be quite different from one knowledge base to another.

The keyword approach reduces the domain of questions that can be answered correctly to the number of questions based on the keywords that the system has. The use of keywords for categorization of a question can add a large amount of complexity to a system. Most systems that use the keyword approach use compromises to allow for good Performance in a domain without too much unneeded Complexity.

4. IMPLEMENTATION

The proposed architecture has been implemented in three main stages Descriptive Database construction, Unstructured Query Filtration and Mapping stage. The system has been tested on Enterprise Employee Database.

The Domain and Language Construct based mapping to convert Natural Language Query to SQL provides the user with an interface to enter query in Natural Language and displaying the converted SQL as shown in "Figure 2". And the result data for correct SQL as shown in "Figure 3".

The database here is constructed using SQLyog (MYSQL) and integrated to the interface using Jdbc and odbc as the whole system is implemented in JAVA environment. We have implemented the Dataset used for mapping between different stages and Query processing using J2SE as follows:

4.1 Dataset Used for Mapping

These datasets are maintained either to represent the actual dataset on which the query is performed or it includes the process dataset that are used to perform different operations of user query. These datasets are defined at different stages of work. These dataset are either defined in the tabular form or the text form. EMPLOYEE (actual data on which the user query in performed), STOPLIST (defined in textual form based on this list filtration on initial stage is done), DOMAIN CONSTRUCT (include the data values that a user can input as query), LANGUAGE CONSTRUCT (includes noun, verb, pronoun, adjectives etc.), RULES (This dataset is defined to arrange the user query under NPL. This will obtain the all the rules possible on the query to organize the query and to generate some meaningful information from it).

4.2 Query Processing

"Figure 2" represents the interface used by the user to input the query. Entered Natural language Query is scanned or filtered for the extraction of keywords based on the STOPLIST dataset. The now formulated sentence is used further by Lexical Analyzer for the identification of LANGUAGE and DOMAIN CONSTRUCTS making use of the dataset and also specific data values are identified. Then the NLP mapping rules embedded in system convert the NLP Query to SQL query. If the entered query is correct as stated by the mapping rules then integrating the Language Constructs (Find, Look, Select, greater than, atleast etc.) Along with the entered Domain Construct (Name, Age, Salary, marital_status etc.) And also by integrating SELECT, FROM, WHERE keywords of SQL the SQL Query is formulated. If the entered query is incorrect then a dialog box with incorrect Construct is prompted.

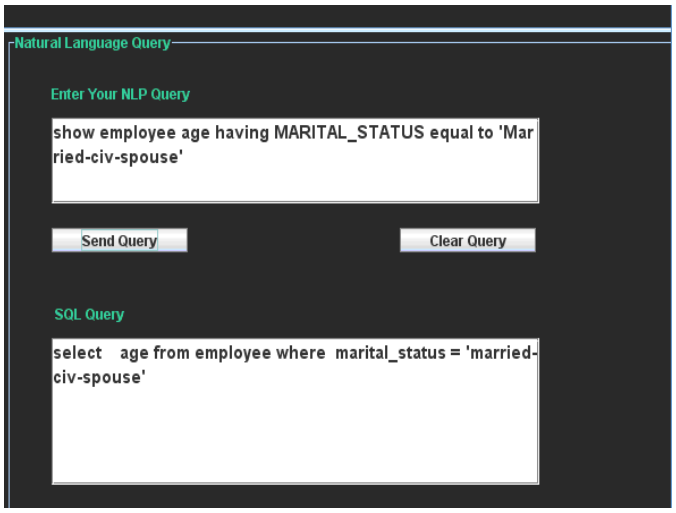

Fig 2: The Interface to enter the User Query.

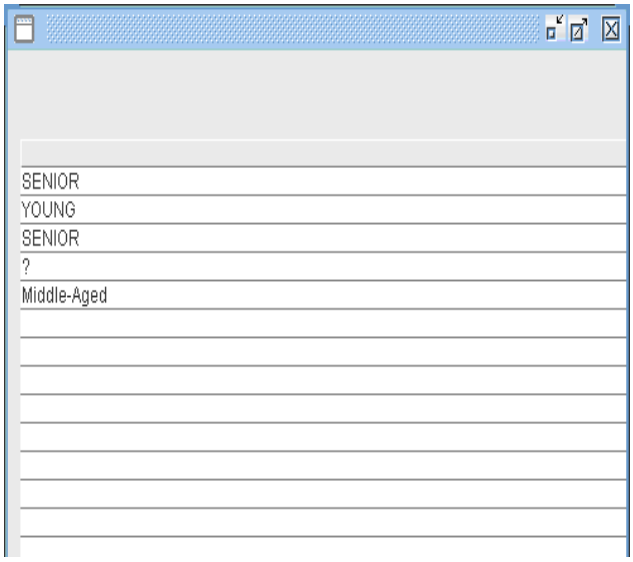

Fig 3: The Result of the correct SQL Query.

5. RESULT AND CONCLUSION

Several Methods and architecture have been adopted for the implementation of various Natural language Interfaces. Keyword based Interface is now also in the same list and several advancements has also been made towards increasing its efficiency. Recently, a combined approach using both NLIDB and KBIDB is developed for increasing the efficiency and accuracy of queries as well as accessing the Database. But this proposed architecture is advancement over the available system with broader range of query handling because of the Query constructs or the special keywords such as atleast, atmost. And also the Domain and Language specific constructs defined during the mapping. The presented work is implemented on employee dataset with the mapping based on query constructs, domain constructs and operational constructs. The work is implemented in java environment and verified on multiple user queries. The obtained results show the effective query conversion and record extraction from the Dataset. The following "Table 1" shows the type of queries inputted to the system and the SQL output formulated by it.

Table 1. Example of Natural Language Query Input and Output generated using System

Natural Language Query (Type of English Sentence)	Generated SQL Result
Show all employees	Select * from employee
Look for employee having salary atleast 3000	Select * from employee where salary >= 3000
Show employee having salary atmost 4000	Select * from employee where salary <= 4000
Show employee having gender equal to female	Select * from employee where gender = 'female'
Show employee age having MARITAL_STATUS equal to 'married-civ-spouse'	Select * age from employee where marital_status = 'married-civ-spouse'
Look for all employees having race white and female	Incorrect Syntax of SQL Query [No Proper Construct Found]
Look for employee age having RELATIONSHIP not equal to 'Husband'	Select * age from employee where relationship <> 'husband'
Search employee with HOURS equal to 'overtime'	Select * from employee where hours = 'overtime'
Look for senior employee education	Incorrect Syntax of SQL Query [No Proper Construct Found]
Find all employee salary with AGE equal to 'SENIOR'	Select * salary from employee where age = 'senior'

6. FUTURE WORK

The presented work is defined specific to a particular domain database. In future, the work can be generalized. It can also be improved by using some optimization approach. The proposed work here is restricted respective to number of language constructs specified. In future, the work can be improved with larger dataset. The work can be implemented on complex queries such as nested query or sub queries. The work is specialized on enterprise employee dataset, in future the work can be implemented on other dataset.

7. REFERENCES

[1] D.J.P.H.N.P. Axita Shah, "*NLKBIDB- Natural Language and Keyword Based Interface to Database*", in IEEE, 2013.

[2] Vesper Owei," *Natural Language Query Filtration in the Conceptual Query Language*",1997 IEEE.

[3] Kunar S Vaisla & Ashish Kumar, "*Natural Language Interface to Database: Development Techniques*", Elixir Comp. Sci. & Engg. , 2013.

[4] A. Kaur and P. Bhatiya, "Punjabi Language Interface to Database", M.Tech dissertation, Department of CSED, Thapar Universirty, 2010.

[5] D.D & Ashish Tamrakar, "*Query Optimization using Natural Language Processing*", Int. J. Tech, Vol. 1, Issue 2, 2011.

[6] D.M.M. D.S.S. Neelu Nihalani, "*Natural Language Interface for Database: A Brief Review*", IJCSI, Vol. 8, Issue 2, 2011.

[7] Y. S. Hu Li, "*A Word Net Based Natural Language Interface to Relational Database*", in IEEE, Wuhan, China, 2010

[8] D.S.S. V.K. N. S. Alok Parlikar, "*NQML: Natural Query Markup Language*", NLP - KE '05, 0-7803-9361-9/05©2005 IEEE

[9] D.S.S. D.M.M. Neelu Nihalani, "*An Intelligent Interface for Relational Databases*", IJSSST, vol. 11, No.1.

[10] N.J.M. Porforio p. Filipe, "*Database and Natural Language Interfaces*", Portugal, 2000

[11] George A. Miller, *"WordNet: A Lexical Database for English*", communication of the ACM, Vol. 38, No. 11, 1995

[12] WordNet WordNet, http:/wordnet.princeton.edu/

[13] The Stanford Parser: A statistical Parser. http://nlp.stanford.edu./software/lex-parser.shtml

[14] Link Parser Grammar link-http://www.link.cs.cmu.edu/link/submit-sentence-4.html

[15] Z. S. Z. A. A. A. I. S. E.-F. Faraj A. EI-Mouadib, "*Gneric Interactive Natural Language Interface to Databases (GINLIDB)*," International journal of computer, vol. 3, no. 3, 2009.

[16] Y. S. Hu Li, "*A Word Net Based Natural Language Interface to Relational Database*", in IEEE, Wuhan, China, 2010

[17] en.wikipedia.org/wiki/Natural_language_processing.

[18] S.B.P. T. D. H. D. T. Nguyen, "*A Vietnamese Natural Language Interface to Database* ", in IEEE, china, 2002.

[19] S. C. D. Sanjay Agrwal, "*DBXplorer:A system for keyword base search over relational database*," in IEEE, 2002.

[20] z. j. y. D. Shang wank, "*NUITS: A novel user interface for efficient Keyword search over database*," in citeseer, China, 2006.