NLP for Information Retrieval using B Trees

Manasamithra P.
Department of Computer Science
JSS Science and Technology University (Formerly SJCE)

H. C. Vijayalakshmi
Department of Computer Science
JSS Science and Technology University (Formerly SJCE)

ABSTRACT
Information retrieval is very important area in any of the IT applications. Systematic data storage is essential in information retrieval. In conventional method, data is stored in a structured format and retrieved using SQL queries which requires technical knowledge. There is a necessity to retrieve the data using natural language. A hybrid system to query and retrieve the data from the database using natural language is implemented. It is a combination of keyword based and semantic analysis methods. M-way B-tree is used to store the keywords which act as knowledge base. Analysis shows that using B-tree is found to be faster and superior in retrieving information.

Keywords
B-tree, Database Management System, Information retrieval, Keyword based search, natural language queries, Semantic Analysis, SQL,

1. INTRODUCTION
Information retrieval is an emerging technology in Information technology field. Every application needs storing and retrieval of application specific data. The traditional way of doing it, is through various SQL queries. This requires high technical knowledge about the usage of the SQL tools and the structure of relevant database schema. It is hard for common people not having technical knowledge to use these kinds of tools. In this regard, Natural Language Processing concept started evolving rapidly. NLP made human-computer interaction possible through human natural language. Application user can query the database in any of the human languages like English and get the relevant answer using NLP techniques.

Several researchers are working in this direction. Many researchers have proposed different ideas and methodologies to convert natural language query to system understandable query such as SQL query. In this paper, we have proposed a method for conversion of natural language query in to SQL query using a hybrid approach. It includes keywords based and semantic based techniques using an efficient data structure to store the knowledge base.

This paper is organized in to 6 sections. Section1 is about brief introduction and section2 is the literature survey about the work carried out by different researchers using NLP techniques. Section 3 provides the overview of the System. Section 4 gives the implementation details. Section 5 tabulates the results of the experiment and does a comparative analysis of result with and without using B-tree. This is followed by Conclusion and Future Work in Section 6.

2. LITERATURE SURVEY
LUNAR is a system which answers about rock sample brought back from the moon. It uses an Augmented Transition Network (ATN) parser and Woods' Procedural Semantics. It has linguistic limitations and was able to handle 78% of users request and this ratio increased to 90% when dictionary errors were corrected. But it was misleading. LIFFER/LADDER used a semantic grammar to query a distributed database. It is based on three layered architecture, Information Natural Language Access to Navy Data (INLAND), Intelligent Data Access (IDA), File Access Management (FAM). RENDEZVOUS focuses on query paraphrasing and in engaging users in clarification dialogs when there is a difficulty in parsing input as mentioned by Axita Shah et al. [5].

PLANES is able to understand and answer poorly framed questions by interacting with the user. CHAT-80 processes the question into three stages such as 1) Representing a word in logical constants, 2) Representing verbs, nouns and adjectives with their prepositions as a predicate and 3) Representing complex phrases or sentences by conjunctions of predicates on the database. It consists of facts about 150 countries of the world and small set of English language vocabulary, implemented in PROLOG [5].

EUFID consists of three major modules namely analyzer module, mapper module and translator module. Word Alignment based semantic parsing (WASP) uses Prolog as the formal query language to build a semantic parser. There is no necessity of prior knowledge of syntax. The System is able to build a semantic parser from annotated corpora which does not require creating a grammar manually. It was evaluated on GEOQUERY domain and achieves 86.14% precision and 75% recall. It was also evaluated on English, Spanish, Japanese and Turkish from which Japanese and Turkish has the lowest recall.

In [1], Fei L et al., used dependency parser to understand the natural language query linguistically. Stanford parser is used to generate dependency parse tree. The system proposed in [2] by Garima Singh et al., is an algorithm based technique to generate the sql query from the natural language. The system analyses Natural language query in series of steps and at each stage the data is further processed to finally form a query leading to its execution.

3. SYSTEM OVERVIEW
Figure 1 depicts the overview of the proposed system. The entire system that we have implemented consists of mainly four parts namely preprocessing, Natural Language Processor, Knowledge Base and Query translator. Each discussed below.

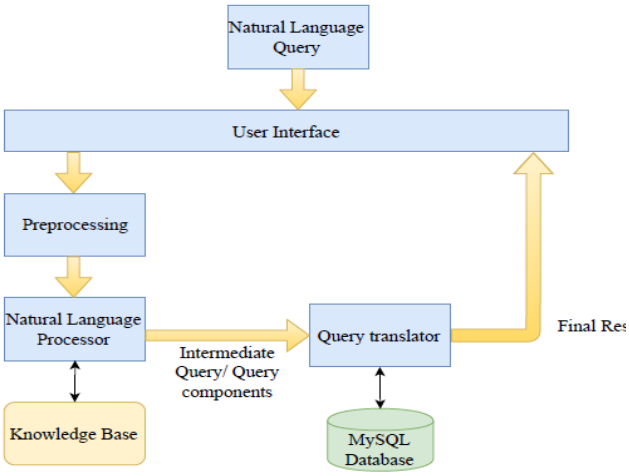

Figure 1. System Overview of Information Retrieval using B-tree

3.1 Preprocessing

Preprocessing is a very important step in NLP. Parsing and cleaning are the most important tasks performed in this phase. User entered query from user interface is the input for this step. The input query undergoes various preprocessing steps. This includes elimination of stop words, changing the case to all lower case and tokenization.

3.2 Natural Language Processor

Natural Language Processor is main part of the system where actual processing takes place. NLP Processor is a combination of Semantic Analysis technique and keyword based approach. Semantic Analysis is a choice when none of the words in the entered query matches with the keywords in the knowledge base. It means that either the query is invalid that it does not fit into the context of database that we have. Or there is a chance that some of the keywords are missing in the knowledge base even though it is a relevant query. If it is first case, Semantic Analysis also cannot produce the result. In case of second option, Semantic analysis will try to form the query using the linguistic relationship between the words from the Natural language query. The output of this step is intermediate query in case of Semantic analyzer and components of the query in case of Keywords based approach. This is done by using Stanford dependency parser. The details of dependency parser are given below.

Stanford Dependency Parser

Many Natural Language processing implementations used Stanford dependency parser as the syntactic and semantic analyzers. Valentin Ilyich Spitkovsky et al. [21] proposed Stanford typed dependencies representation which is designed to provide a simple description of the grammatical relationships in a sentence that can easily be understood and effectively used by people without linguistic expertise who want to extract textual relations by. Marie-Catherine de Marneffe in [22] mentioned that, rather than the phrase structure representations that have long dominated in the computational linguistic community, it represents all sentences relationships uniformly as typed dependency relations which called 'typed dependency parser'. Based on the grammatical relationships, it is possible to determine the component of the SQL query.

Example: Tree representation for the query - Return all the employees who have salary more than 20000

The tree representation given below is based on the linguistic relationship between the words. Here "all" acts as adjective for the noun "Employees". So, from this we can determine that Employee as a table name and "all" as a keywords which is to retrieve all the records of Employee table.

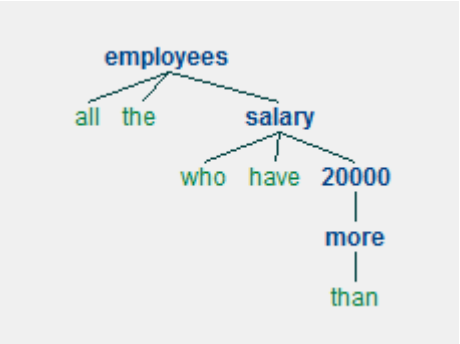

Figure 2. Stanford dependency parse tree

3.3 Knowledge Base

The efficiency of the system will be increased as we use more robust data structure to store the keywords. All the possible keywords should be pre-stored in case of keyword based approach. Storing and searching the keywords in a flat file is expensive. This will increase the search time as the searching method is sequential. Here, we have used a well-known data structure called B-tree. All the keyword combinations are stored in different m-way B-trees with a root node and all the leaves in alphabetical order so that search becomes easy. Separate B-tree is constructed for table names, attributes, constants; escape characters/stop words and so on. A sample 4-way B-tree for all the escape characters is as shown in Figure 3.

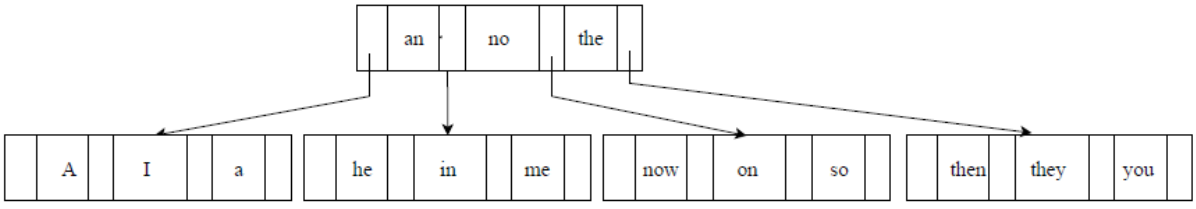

Figure 3. Sample B-tree for storing escape characters

3.4 Query Translator

The result of comparison of tokenized keywords with the knowledge base is the input for this stage. Based on the comparison, tokens are classified as table name, attribute names or constants. Find out the type of query from the classification result. i.e. the query with "where" clause or query with "group by" etc. Based on the type of query, keywords are placed in

appropriate position to form the query. This is followed by query execution in MySQL to get the actual result.

4. IMPLEMENTATION DETAILS

This section explains how exactly the system is implemented in detail. We have used Python for this implementation along with dJango web interface for UI implementation. Detailed system structure is as shown in Figure 4.

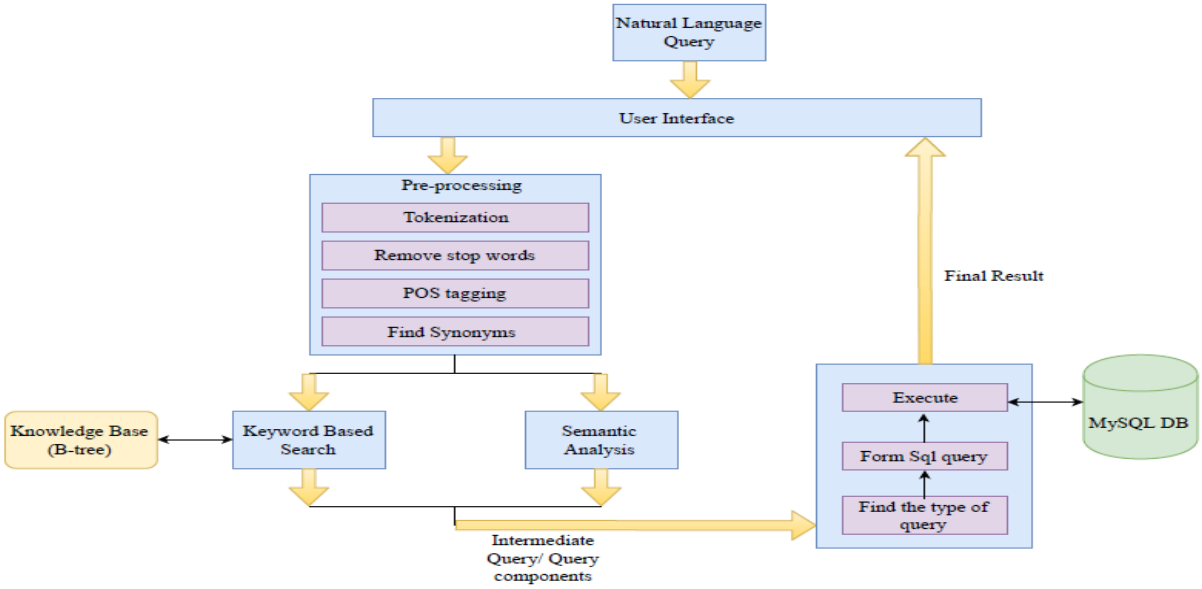

Figure 4. Detailed System structure used for Information Retrieval using B-Trees

User interface: Our application provides a user-friendly interface to the end user who can provide his/her query in English and Submit. Just a single click does everything for user.

Tokenization: Once the query is entered, next step is to divide them into set of atomic tokens which cannot be broken further. In the keyword based approach, this step is carried out by storing space separated words given in the sentence and maintaining it in a list. In the semantic analysis method, tokenization is part of dependency parsing.

Remove Stop words/Escape characters: Once the tokens are generated, unwanted characters should be removed. These are the words which do not contribute anything in forming SQL query. For e.g., 'a', 'an', 'the' etc are some of the stop word characters. The training data is stored previously and compared with the list of tokens extracted from the natural language query. Matched ones are removed from the list.

Part of Speech Tagging: This is done by using POS library available in the python. In case of Semantic analysis method, Stanford parser gives typed dependency parser with each word attached with their part of speech notation. This is very important in Natural Language Processing to determine the relationship between the words. Also, POS tagging helps in routing the search to appropriate B-tree.

Keyword based Approach:
After the common steps mentioned above, next step is to compare each token with the different categories of training data. The training data would be stored in different B-trees. One for escape characters, one for table names, attributes etc. For the comparative analysis purpose, we have also implemented the keywords based system by storing training data set in the flat files. Each time when we compare, file operation is carried out and read from them sequentially.

But this is time consuming when we have very large set of data. B-tree makes this task easier by storing them in alphabetical order. By using tree parsing algorithms, we can easily carry out the comparison process. B-tree is implemented in such a way that we can construct the b-tree of any order based on the number of keywords to be stored. Any new keywords to be added will take a place in an alphabetical order. Searching takes place by finding out the path to the keywords to be searched. This reduces the time by making the search only in either left or right of the root it belongs to recursively. The system will proceed to process the query only if database or table name given in the natural language query exists in the knowledge base.

Semantic Analysis:
Semantic analysis is carried out by using dependency parser. This is helpful when the keywords stored do not yield good result. The linguistic dependency between the words in the sentence is captured using dependency parser. Stanford dependency parser is used for this purpose. This helps the system to understand the query linguistically. Each word in the natural language query acts as node and the edge between any words is the linguistic dependency relationship between the tokens. The edge between the tokens is determined based on the part of speech. Based on the relation, determine the place of each word in the query.

Query Translator:
The tokens are classified into tables, attributes, constants and other database entities based on comparison match. Find out the type of query once the classification of the tokens is determined. System supports simple queries, query containing 'where' clause, join queries with multiple tables, group by, having, query with constant values in the where clause, one level nested queries and queries with multiple conditions in the where clause. Based on the type of the query, keywords are placed in appropriate place to form the query.

We have experimented on ten different databases like Student, Employee, Hospital, University etc. One of the databases used

for the implementation is the Employee database and the Schema diagram is as shown in Figure 5.

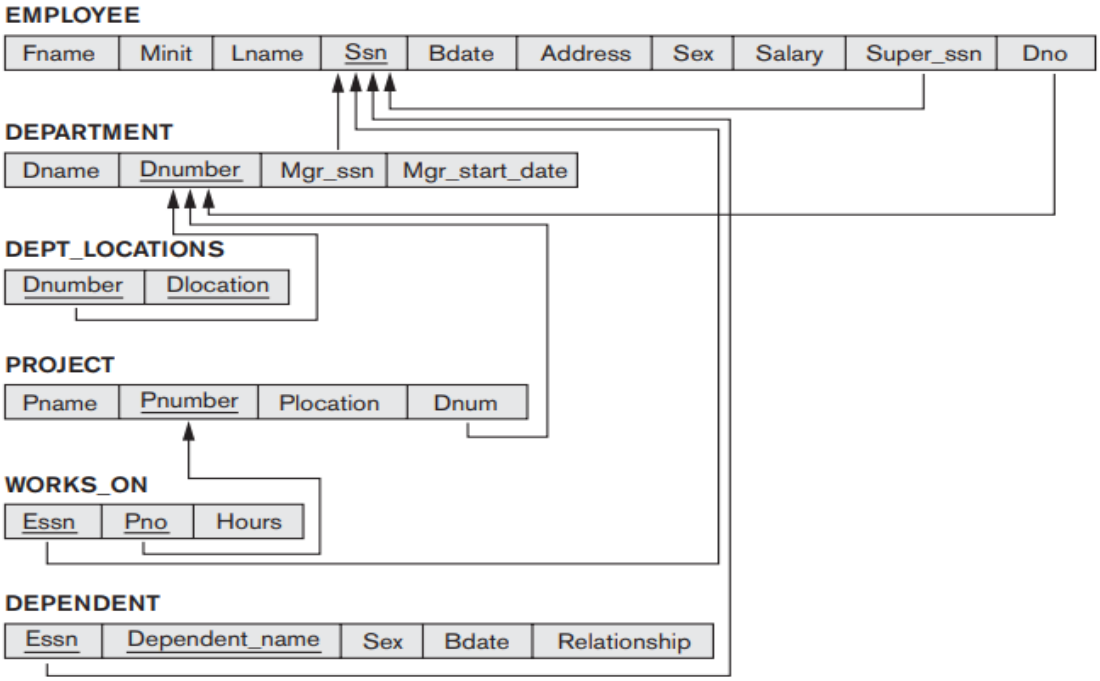

Figure 5. Schema diagram of Employee database

5. RESULT AND ANALYSIS

A set of 10 databases, 500 table names, 1500 attributes are used for testing. This implementation resulted in good accuracy as a combined approach. All types of queries including one level of nesting are possible with this system. A comparative analysis is done by carrying out the keyword based search sequentially and by storing the keywords in B-tree. Since, we have also adapted a technique of storing the knowledge base in an efficient data structure called B-tree; it also enhanced the efficiency of the system, hence reduced execution time. Some of the natural language queries and corresponding SQL queries are as shown in Table 1.

We have made a comparative analysis by using the B-tree as knowledge base versus flat files. There is a huge difference in the execution time. The result is tabulated in table 1. As we can observer from the Table 1, Execution time reduced almost by 86%. It also helps in enhancing the space efficiency.

Table 1. Result of comparative analysis

Sl No	Natural Language Query	SQL query	Execution Time using B-Tree(in Sec)	Execution Time without using B-Tree(in Sec)
1	Get all the Employee details	Select * from EMPLOYEE;	0.6783	4.8234
2.	Return all the Employee details who has salary more than 30000	select * from EMPLOYEE where Salary>30000;	0.573	1.59
3.	Give me all the employee names who works in same department location	Select Fname, Lname from Employee, DEPARTMENT, DEPT_LOCATIONS where Employee.Ssn = DEPARTMENT.Dnumber and DEPARTMENT.Dnumber = DEPT_LOCATIONS.Dnumber group by DEPT_LOCATIONS.Dlocation;	0.69	2.258
4.	List the Project names which belongs to HR Department	Select Pname from PROJECT, DEPARTMENT where PROJECT.Dnum = DEPARTMENT.Dnumber and	0.759	1.987

		DEPARTMENT.Dname = “HR”;		
5.	Count the total number of employees in the organization.	Select count(*) from EMPLOYEES;	0.568	0.987
6.	Give me details of employee whose name is “John”	Select * from EMPLOYEE where Fname=”John”;	0.305	1.508
7.	Return the details of all Female Employees	Select * from EMPLOYEE where Sex=”Female”;	0.496	3.456
8.	List all the project names which belongs to same department	Select Pname from PROJECT group by Dnum;	0.509	2.964
9.	Select all the unique data from Department and Department location.	Select * from DEPARTMENT NATURAL JOIN DEPT_LOCATION;	0.578	3.237
10.	Return the unique data from Employee and Department	Select * from EMPLOYEE NATURAL JOIN DEPARTMENT;	0.435	3.572

6. CONCLUSION AND FUTURE ENHANCEMENT

This system is implemented by a combined approach of both keyword based and semantic analysis which yielded a good result. Many researchers have used sequential search method for keyword based mechanism. Few of them used XML knowledge base as it is a structured mechanism to retrieve the keywords and other related information. Here, we have used m-way B-tree data structure as knowledge base. This enhances the search burden which was present in sequential search. When there is a huge amount of data in the database, this method is very useful as the result is very quick. For very large database, keyword based mechanism takes lot of time for search in case of sequential. But, the use of efficient data structure enhances the search burden on the algorithm.

This system is designed as a proof of concept to demonstrate the search time reduction in case of m-way b-tree usage in Information retrieval. So, it is restricted to certain types of queries as mentioned in Table 1. The system can be enhanced to include much more types of query such as all types of join, multiple levels of nesting etc.

7. REFERENCES

[1] Fei L, H. V. Jagadish, “Constructing an Interactive Natural Language Interface for Relational Databases”, Proceedings of the VLDB Endowment, Vol. 8, No. 1, 2014

[2] Garima Singh, Arun Solanki, “An algorithm to transform natural language into SQL queries for relational databases”, Selforganizology, 2016

[3] Subhabrata Sengupta, Prasun Kanti Ghosh, Saparja Dey “Automatic SQL Query Formation from Natural Language Query”, July 2014

[4] Prabhdeep Kaur, Shruthi J, “ CONVERSION OF NATURAL LANGUAGE QUERY TO SQL”, International Journal of Engineering Sciences & Emerging Technologies, Jan. 2016.

[5] Axita Shah, Dr. Jyoti Pareek, Hemal Patel, Namrata Panchal, “NLKBIDB - Natural Language and Keyword Based Interface to Database”, International Conference on Advances in Computing, Communications and Informatics (ICACCI) IEEE , 2013

[6] Filbert Reinaldha, Tricya E. Widagdo, S.T., M.Sc., “Natural Language Interfaces to Database (NLIDB):Question Handling and Unit Conversion”, 2014 IEEE

[7] Azilawati Azizan, Zainab Abu, Shahrul Azman Noah, “Query Reformulation Using Ontology and Keyword for Durian Web Search”, 2016 Third International Conference on Information Retrieval and Knowledge Management

[8] Manavalan, Subrata Chattopadhyay, Mangala, Prahlada Rao, Sarat Chandra Babu, Akhil Kulkarni, “Experiments on Information Retrieval Mechanisms for Distributed Biodiversity Databases Environment”, IC3I,IEEE, 2014

[9] Sanket S.Pawar, Abhijeet Manepatil, Aniket Kadam, Prajakta Jagtap, “Keyword Search in Information Retrieval and Relational Database System: Two Class View”, ICEEOT, 2016

[10] Xuan Xuan, Liu Jianbo, Yang Jin, “Research on the natural language querying for remote sensing databases”, International Conference on Computer Science and Service System, 2012

[11] Xu Yiqiu Wang Liwei Yan Shi, “The Study on Natural Language Interface of Relational Databases”, 2nd Conference on Environmental Science and Information Application Technology, 2010

[12] Mahesh P.Gaikwad, Natural Language Interface to Database, International Journal of Engineering and Innovative Technology (IJEIT) Volume 2, Issue 8, February 2013

[13] Pooja A.Dhomne, Sheetal R.Gajbhiye, Tejaswini S.Warambhe, Vaishali B.Bhagat, “ACCESSING DATABASE USING NLP”, IJRET, Dec-2013

[14] Avinash J. Agrawal, Dr. O. G. Kakde, “Semantic Analysis of Natural Language Queries Using Domain Ontology for Information Access from Database”, I.J. Intelligent Systems and Applications, 2013

[15] Rukshan Alexander, Prashanthi Rukshan,Sinnathamby Mahesan, “Natural Language Web Interface for Database (NLWIDB)”, Proceedings of the Third International Symposium, SEUSL: 6-7 July 2013

[16] Gaganpreet Kaur, "Usage Of Regular Expressions In Nlp", IJRET, Jan-2014

[17] Akshay G. Satav, Archana B. Ausekar, Radhika M. Bihani, Mr Abid Shaikh," A Proposed Natural Language Query Processing System", International Journal of Science and Applied Information Technology, April 2014

[18] K. Javubar Sathick, A. Jaya, "Natural language to SQL Generation for Semantic Knowledge Extraction in Social Web Sources", January 2015

[19] Rongrong Zhang, Qingtian Zeng, Sen Feng, "Data Query Using Short Domain Question in Natural Language", 2010 IEEE

[20] Johanna Monti, Mario Monteleone, Maria Pia di Buono, Federica Marano, "Natural Language Processing and Big Data An Ontology-Based Approach for Cross-Lingual Information Retrieval", IEEE, 2013

[21] Marie-Catherine de Marneffe and Christopher D. Manning, "Stanford typed dependencies manual" September 2008.

[22] Valentin Ilyich Spitkovsky, "Grammar Induction And Parsing With Dependency-And-Boundary Models", December 2013

[23] http://www.nltk.org