International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395 -0056
Volume: 04 Issue: 04 | Apr -2017 www.irjet.net p-ISSN: 2395-0072

Knowledge Extraction from Database using Natural Language Processing

A.R.FALLE[1], S.K.PANHALKAR[2], A.P.JADHAV[3], K.V.KAMBLE[4], A.A.SALUNKHE[5], D.V. MIRAJKAR[6]

[12345]Student, Dept. of Computer Science and Engineering, D.I.E.T college, Maharashtra, India
[6]Professor, Dept. of Computer Science and Engineering, D.I.E.T college, Maharashtra, India

---

Abstract - *Natural language processing is a field of computer science concerned with the interactions between computers and human (natural) languages. Natural language processing (NLP) is the ability of a computer program to analyse natural language & speech. NLP is a component of artificial intelligence (AI) which is used to understanding complex language. The main purpose of Natural Language Query Processing is for an English sentence to be understood by the computer and appropriate action taken. Here, in this paper we convert the user question in English sentence which is natural language into a SQL query and that query is executed to get back results of that query entered*

Key Words: **Natural Language Processing, NLP**

1.INTRODUCTION

The field of study that focuses on the interactions between human language and computers is called Natural Language Processing. Natural language processing (NLP) is one of the most important technologies of the information age. NLP allows a machine to understand natural language. NLP is a way for computers to analyze, understand, and derive meaning from human language in a smart and useful way.

The applications that will be possible when computer would be able to process Natural Language translating language accurately and in real time, or extracting and summarizing information from a variety of data sources depending on the user's request.

NLP systems capture meaning from an input of words (sentences, paragraphs, pages etc.) in the form of a structured output. Language processing is most essential part of our system. This system is processing the natural language which is English entered by the user. With that input our system predicted the suitable result of that sentence.

This article help for the use the dynamic database for the apply the query which get from the system. This system also helpful for non technical person who has only knowledge about the English then that person is able to handle that system easily find result of his/her question without knowledge of sql query.

2. LITERATURE SURVEY

Miguel Liopis, Antonio Ferrandeza in [1] explained that NLIDB system and classify them based on the different approaches that they implement.

Laszlo Kovacs in "SQL Generation for Natural Language Interface" [2] , uses the Automata to process the query and implementation of Query Processor using automata and Natural language processing.

Androutsopoulos, G.D. Ritchie, P. Thanisch.[3] in "Natural Language Interfaces to Databases – An Introduction" ,describes hints about the capabilities of existing NLIDB, it does not contain complete descriptions of particular systems, nor is the purpose of this paper to compare particular NLIDB.

Fei Li, H.V.Jagadish. [4] in "Constructing an interactive Natural Language Interface for Relational Database." This paper uses the dependency parser, Parse tree node, parse tree and use the My SQL as the RDBMS, and the NLP parser as the dependency parser.

The intellectual contribution of this paper are as follows:

I. Interactive query mechanism: they have design an interaction mechanism for NLIDB to enable users to ask complex queries and have them interpreted correctly.

II. Query tree: they have design a query tree structure to represent the interpretation of a natural language query from the database's perspective

Alessandra Giordani and Alessandro Moschitti. in [5] "Semantic Mapping between Natural Language Questions and SQL Queries via Syntactic Pairing" this paper, they exploit mapping at syntactic level between the two

languages and apply machine learning algorithms to derive the shared shallow semantics.

As we outlined in the previous section, there have been only one approach to the construction of natural language processing system. The only approach of that system is that they only support the static database or the database provided them by programmer. They don't support any change into that database. It was the serious limitation of it.

In this article we will explore the system which supports the dynamic database of user to access it for different person of different organization. As others paper doesn't include in their system.

In the next section, we will explore the each of these approaches. It is important to emphasize that we do not claim one of these approaches to be better than others, as each of the approaches has its advantage and disadvantages. However, we will evaluate the convenience of each of these approaches in regards to the main goal of our research work: Knowledge Extraction from Database using NLP.

3. METHODOLOGY

The main improvement in our proposal compared to the other existing system is to support the Dynamic databases of user. This system supports the partial change or all changes in database of the end user. The previous system WASP (Word Alignment-based Semantic Parsing) is a system developed at the University of Texas by Yuk Wah Wong. This system was designed to get "complete, formal, symbolic, meaningful, representation of natural language sentence". The main drawback in this system is that it only parses given natural language input string and generates query, but could not execute query and take appropriate actions. This makes database untouched.

Another improvement in our system is that it not only generates query but also provides user expected result.

Steps involving to implement system is below

1. **Preprocessing**
   - **a)white space removing**
   - **b)comparative replacement**
   - **c)negation processing**
   - **d)lemmatize processing**
   - **e)special word recognition**
2. **Entity Recognition**
3. **Query Generation**
4. **Result**

Chart-1: Steps of Algorithm

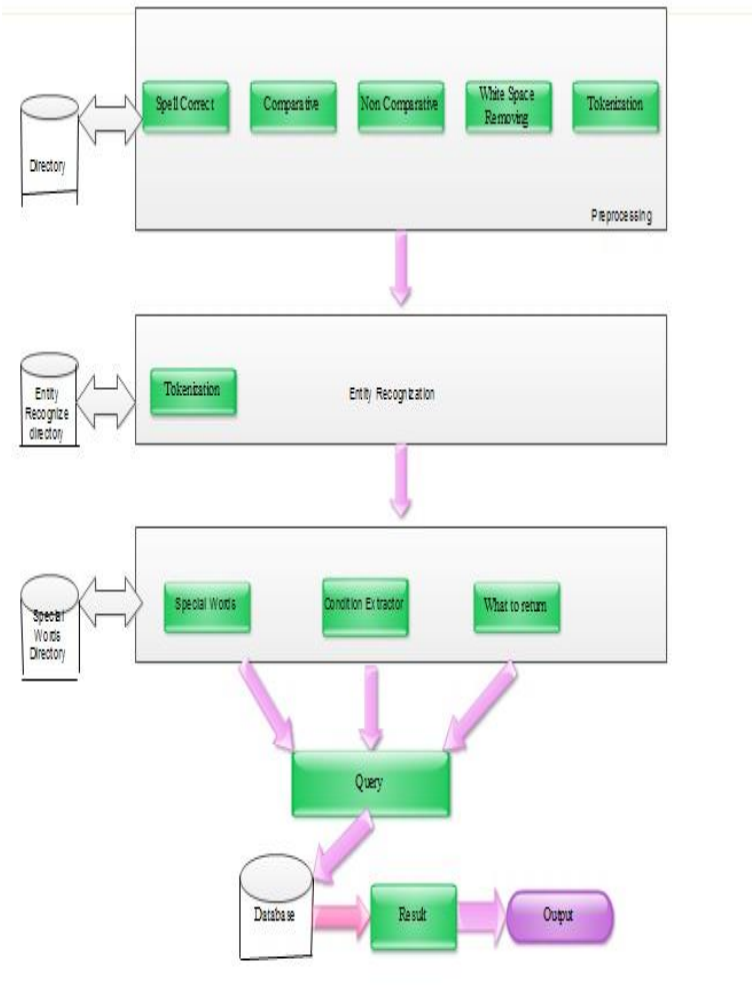

Fig- 1: System Architecture

4. SYSTEM MODULES

4.1 Pre-Processing-

Language pre-processing is eliminating unwanted, meaningless contents from user string. Pre-processing module removes the extra white spaces from input. Pre-processing means converting user input string into at least proper[1] sentence. Grammatical mistakes, spelling mistakes are also handled in this module. This approach introduces us to functions like Auto-correct, Auto-predict, comparative replacement. Comparative replacement is replacing words with their associated notations.

(a) White Space Removing- White space remove function removes extra whitespaces from user input. White spaces have no meaning in language processing and hence are removed for further processing.

(b) Comparative Replacement- User when enter the natural query(question) as an input then there is a possibility that he might use special symbols as well as text (e.g. greater than, higher than, bigger than this word are replace with proper symbol for pass through next module for SQL query generation).

(c) Negation Processing- Negation processing function replaces negation conditions with their appropriate conditions. E.g. "Not less than" replaced by "greater than".

(d) Lemmatize Processing- Lemmatize processing function removes suffix or prefix and coverts the word into its original form. E.g. "Running" converted to "Run", "failed" converted to "fail".

(e) **Special Words Recognition**- In this function user query is spitted to get list of separate words. This list is then compared with the special words already defined in dictionary.

4.2 Entity Reorganization:

Named entity recognition (NER) also known as entity identification, entity chunking and entity extraction is a subtask of information extraction that seeks to locate and classify named entities in text into predefined categories such as the names of person, organization, address of employee, marks of students etc. Here entity refers to candidate words listed out from the given user input string.

The entity recognition function helps to system to retrieve actual information from database. Entity recognition is used relate the candidate words with their column names. These column names of the candidate words helps in describing the information user want

4.3 Query Generation-

Query generation is generating the SQL query using user input for accessing information. Example - user question ;- Tell me the names of students having marks less than 60.

Query generation mainly includes two parts:-

1) What to return - what to return is finding out actually what user wants as a output. In above example "names of students" is what to return.

2) Conditions - In above example "marks less than 60" is condition. Using this condition user will be able to find out students having marks less than 60.

Query is generated combining these two parts: SELECT (What to return) FROM DATABASE_NAME WHERE (Condition);

4.4 Result Post Processing-

Generated query is fired on database, resultant output is given to user and stored onto buffer for further use. Buffers are small size memory used to store and retrieve data fast. In case, user requests same information then buffer comes into frame. Results are directly given to user from buffer without processing input string.

4.5 Regex for extracting condition-

Following are some regex used in system for extracting condition -

col_([a-z]*)(is|are|\s)+ope_(>|<|>=|<=|=|!|!=)\s(value_(\w*\s?))

col_([a-z]*)(\s|is|are)+(value_(\w*))+

col_([a-z]*)(\s|is|are)+values_list-((value_(\w*_\s?))+)+

col_([a-z]*)(\s|is|are)+ope_(>|<|>=|<=|=|!|!=)values_list-((value_(\w*_\s?))+)+

5. CONCLUSION

This system describes an interactive natural language query interface for relational databases. Given a natural language query, system first translates question to a SQL statement and then evaluate it against an RDBMS. To achieve high reliability, our system explains to the user how query is actually processed.

REFERENCES

[1] Llopis, Miguel, and Antonio Ferrández. "How to make a natural language interface to query databases accessible to everyone: An example." *Computer Standards & Interfaces* 35.5 (2013): 470-481

[2] Kovács, László. "SQL generation for natural language interface." *Computer Technology and Computer Programming: Research and Strategies*. Apple Academic Press, 2011. 90-98.

[3] Androutsopoulos, Ion, Graeme D. Ritchie, and Peter Thanisch. "Natural language interfaces to databases–an introduction." *Natural language engineering* 1.01 (1995): 29-81.

[4] Li, Fei, and H. V. Jagadish. "Constructing an interactive natural language interface for relational databases." Proceedings of the VLDB Endowment 8.1 (2014): 73-84.

[5] Giordani, Alessandra, and Alessandro Moschitti. "Semantic mapping between natural language questions and SQL queries via syntactic pairing." *International Conference on Application of Natural Language to Information Systems*. Springer Berlin Heidelberg, 2009.