An intelligent system for transforming natural language queries into SQL and its execution.

Swati Kale 1, Vinay Gahse 2, Piyush Mahatkar 3, Ruhit Ratnaparkhi 4, Nakul Dhawale 5 and Tanay Totade 6

1 2 3 4 5 6 Department of Information Technology, YCCE, Nagpur

Email: s12_kale@ycce.in

Abstract. In today’s world information storing and retrieval plays an important role. Database systems play a key role in the new commercial system for information storage. For accessing the data from database, a person who is not having the knowledge of SQL may find themselves handicapped while dealing with the database. This presents a significant limitation in a developing country such as India, because even today, a very large majority of the population does not have the technical knowledge of how to deal with database systems. In such a case, Natural language processing (NLP) assumes a significant job. NLP is getting perhaps the most dynamic strategies utilized in Human-PC Interaction. It is a part of AI which is utilized for Information Retrieval, Machine interpretation, and Language Analysis. This paper, proposes a system which allows the user to query the database in a compatible mode language, through a convenient Graphical User Interface which results in the data required by the user.

Keywords: Natural language processing, Database, Artificial intelligence, Information retrieval.

1. Introduction:

Conventionally everyone is more comfortable in the language which they speak, writes and, understands. The official language of India is English (UK). Now a day’s most of the people in India speak/understand English but they are not acquainted with technical language like SQL. A programmer or a naïve user can interact with the database system.[1] Natural language processing provides the platform through which not only programmer but every user can communicate with the system without having the knowledge of the technical language. To extract the data from a database management system like MS access, Oracle user must have knowledge of structured query language. To understand this consider the database of library books. Suppose the user wants to access the information for books of a particular author. For retrieving this type of information in oracle user needs to fire a query select * from library where author_name="xyz"; The role of an intelligent system comes here. An intelligent system is developed for the people to interact with database using simple English statements. Using this system instead of remembering all DDL and DML commands it is possible for a user to ask the required data using simple English statement like give me the name of books whose author is xyz. A keen layer is planned which acknowledges basic client's basic sentences as information and converts them into standard SQL questions to recover information from social data sets dependent on an information base. This sort of cooperation is frequently valuable to the individuals who are not experts in informatics; they are intrigued uniquely with regards to looking into information. They might be chiefs or experts, or a person's getting to the information base.
2. Related work:
A lot of research has been done and still going on in developing new methods of natural language interface to database. LUNAR systems introduce in 1973[2] which is based on Augmented Transition Network(ATN). This system basically developed for handling the queries on samples of rock which is brought back from the moon. One of the best data processing system comes in 1978 a LIFTER/LADDAR [3] which stores the information for US navy ships. To parse the input of user queries in NLP a semantic grammar was used in this system. A system which interacts the user when it does not understand the user queries while parsing was introduces in 1977 RENDEZVOUS [4]. A system gives emphases on query paraphrasing. CHAT-80(1980) is nominated for one of the best interaction system with the help of natural language in 80’s. The system makes the use of prolog language. In 2009 English Wizard was introduced which is another tool for translating natural language English database requests into SQL. It was one of the efficient software that converts simple requests of English database into SQL [5][6]. In all above the system interaction language is English. Some of the researchers had also worked for Indian language. In 2014 a system for those people who are comfortably work with Hindi language was introduce. This system is works on rules. The output of the query is in Hindi language [7]. Apart from above ASK, TEAM, PHILIQA, PRECISE are also NLP query translator systems. The author developed the system for accessing the data for railway reservation system by inputting the query in natural language. They obtained 98.89% accuracy [15]. A main purpose of this system proposed by [16] is to improve the procedure of SQL query generation from software specifications by solving the ambiguity problem of natural languages that works on Domain Specific Query Generation. Stanford parser is used for generation of SQL Queries. The system proposed by [17] works on voice based natural language processing Speech recognition technique. User speech is given as input and then converted into text and then text to SQL query and result is displayed in tabular format.

3. Proposed system:
The proposed system is based on parsing and semantic check. The system accepts user’s natural language sentences as input, parses them semantically and builds an SQL query for the database. The functionality is based on the semantics and rules, which can be modified by the system administrator. The system is composed of two modules: a Training system and a Query Building system. Fig 1 gives the architecture of the system.
Modules involved are as follows:

1. **Natural Language Query**: The user is supposed to give the input in Natural Language i.e. English. The input may be any kind the user wish to enter as to get the desired results from the System. The input is typed by the non-technical user over the textbox. The input format contains some convenience rules or format to be followed by user specified by the instruction page.

2. **Tokenisation**: The pre-processing step is Tokenization. In this part of system input query text is converted into unigram tokens. After generation of tokens it is given as input for future processing. These tokens are stored for analysis purpose.

3. **Parsing and Semantic Check**: The tokenisation Process provides with tokens for analysis. The tokens are parsed similar to lexical analysis in compiler designing. Here tokens are checked for semantics of the statement. The excess and non-meaningful words are discarded. The remaining tokens are matched to follow the semantic rules and grammar.

4. **Query Building**: This module focuses on the construction of the query in according to output of parser and rule base. Expressions are evaluated, tables are selected and clauses are analysed. Later the corresponding transformations are done to form SQL query.

5. **Error Log**: Error log keeps records of all the failed attempts done over the UI screen. The input sentences are stored in a text file, so that any technical user can update the data dictionary, helping users to successfully access data in future.

6. **Updating through SQL**: The technical users or administrator can login to look at the failed attempts done by users, which was stored in a text file. With the help of those, he can update the data dictionary. This will help to modify the system, and reduce the failed attempts.

7. **Query Mapping**: The built query is then mapped for the type of operation specified and is mapped over the existing query base or domain. The mapped query is then triggered over the RDBMS and output is displayed in form of tables or human readable format.
8. **Database**: Formally, a database refers to a set of related data and the way it is structured or organized.

9. **GUI Display**: GUI display is the user interface over which the user interacts. It will enhance the efficiency and ease of use. It also displays the output of input query.

10. **Login Screen**: It requires a connection to database to access data. The login screen asks for the user names, database name and its password, as the data is confidential in its own. Unsuccessful attempt results into closing of the application.

11. **Word Check and excess word remover**: It checks all the words in the user question against the data dictionary for its existence. It removes excessive words from the question string.

4. **Methodology**

   In the first step user has to give the input a query in a simple English statement like “what is the salary of Rajiv”. Then a sentence is broken down into tokens like what, is, the, salary, of, Rajiv. The removal of stop words is done. Each token is analyzed into their components, and tags are assigned to each token like a noun, adjective, adverb etc. Before that suffix's are removed and syntactic analysis is also done to check the grammatical mistakes. A limited data dictionary is also used to store all related words about the system. After this, each individual word is mapped into appropriate objects in the knowledge base or database and the meanings of the individual words combine with each other and find out the meaning of a simple English query. Example: Meaning of query: Salary of employee with name Rajiv. A translator will change the given English sentence with SQL query and the result of the same gets displayed.

   SQL Query: Select Salary from emp where emp_name = "Rajiv".

5. **Algorithm**

   5.1. **Query Building Algorithm**:
   1) Accept String
   2) Tokenization of String
   3) Parsing String from left to right for the first time
   4) Match tokens for
      A) Keywords
      B) Table names
      C) End of statement
   5) For each keyword append with empty string 's'
   6) Parsing for second time from left to right,
      A) Searching for attributes
      B) Join or'.' operator statements
   7) String Updating 's' according to Step 6.
   8) Query triggering over database.

   5.2. **Dictionary Update Algorithm**:
   1) Login to admin account
   2) Check Error Log file
   3) Select Operation amongst
      a) Insert
      b) Modify
      c) Delete,
   on the “Word Dictionary” table
   4) Logout
6. Result and Conclusion:

To test the system we have used the data set of student and employee. A question set of 30 queries are used. The performance factor we have used are TP(True Positive), FP(False Positive), TN(True Negative), FN(False Negative) and recall.

| Actual True | Predicted True | Predicted False |
|-------------|----------------|-----------------|
| FN=3        | TP=42          |
| Actual False| TN=3           | FP=2            |

Table 1: Confusion Matrix for 50 NLQ.

Based on the formulas we have calculated Recall, Accuracy, Precision, Error rate as:

Recall = \frac{TP}{TP+TN}

Accuracy = \frac{TP+TN}{(Total \ No \ queries)}

Error rate = \frac{FP+FN}{(Total \ No \ of \ queries)}

Precision = \frac{TP}{(TP+FP)}

| Recall | Accuracy | Error rate | Precision |
|--------|----------|------------|-----------|
| 88%    | 83.33%   | 16.66%     | 91%       |

All the possible combinations of simple queries are tested. The system works for simple aggregate operations. For complex queries, nested queries the system needs an upgradation. This proposed system can be further extended which can be worked for Indian languages like Hindi, Marathi where user is able to give the query input in Indian language.
7. References:

[1] Sreenivasulu M (2014) Information retrieval using natural language interfaces. Int J Comput Appl 92(12):34–37.

[2] Woods WA (1973) Progress in natural language understanding: an application to lunar geology. In: Proceedings of the
(June 4–8, 1973), national computer conference and exposition, ACM. p 441–450 9.

[3] Sathick KJ, Jaya A (2015) Natural language to SQL generation for semantic knowledge extraction in social web
sources. Indian J Sci Technol 8(1):1–10 10.

[4] Hendrix GG, Sacerdoti ED, Sagalowicz D, Slocum J (1978) Developing a natural language interface to complex data.
ACM Trans Datab Syst (TODS) 3(2):105–147 11.

[5] Dhomne PA, Gajbhiye SR, Warambhe TS, Bhagat VB (2013) Accessing database using NLP. IJRET 02(12):589–594
12. Karande ND, Patil GA (2009) Natural language database interface for selection of data using grammar and
parsing. World Acad Sci Eng Tech 3:11–26

[6] Arun Solanki , Ashutosh Kumar A system to transform natural language queries into SQL queries Int. j. inf. tecnol.
https://doi.org/10.1007/s41870-018-0095-2

[7] Sontakke AR, Pimpalkar A (2014) A rule based graphical user interface to relational database using NLP. Int J Sci Eng
Res 3(4):81–84

[8] Woods WA (1973) Progress in natural language understanding: an application to lunar geology. In: Proceedings of the
(June 4–8, 1973), national computer conference and exposition, ACM. p 441–450 9. [9] Sathick KJ, Jaya A (2015)
Natural language to SQL generation for semantic knowledge extraction in social web sources. Indian J Sci Technol
8(1):1–10 10.

[10] Hendrix GG, Sacerdoti ED, Sagalowicz D, Slocum J (1978) Developing a natural language interface to complex data.
ACM Trans Datab Syst (TODS) 3(2):105–147

[11] Dhomne PA, Gajbhiye SR, Warambhe TS, Bhagat VB (2013) Accessing database using NLP. IJRET 02(12):589–594
12. Karande ND, Patil GA (2009) Natural language database interface for selection of data using grammar and
parsing. World Acad Sci Eng Tech 3:11–26

[12] Nihalani N, Motwani M, Silaka S (2011) Natural language interface to database using semantic matching. Int J
Comput Appl 31(11):0975–8887

[13] Giordani A, Moschitti A (2012) Translating questions to SQL queries with generative parsers discriminatively
reranked. In COLING Posters p 401–410

[14] Kaur J, Chauhan B, Korepal JK (2013) Implementation of query processor using automata and natural language
processing. Int J Sci Res Publ 3(5):1–5

[15] Uma M, sheha V Sneha G, Bhavana J, Bharathi B Formation of SQL from Natural Language Query using NLP Second International Conference on Computational Intelligence in Data Science (ICCID-2019).

[16] Anum Iftikhar, Erum Iftikhar (2016) Doma in Specific Query Generation from Natural Language Text IEEE 2016
Sixth International Conference on Innovative Computing Technology (INTECH)

[17] Puja Munde,Sayali Tambe,Afreen Shaikh,Pratiksha Sawant ,Prof.Deepa Mahajan Voice base natural language query
processing International Research Journal of Engineering and Technology(IRJET)e-ISSN: 2395-0056Volume: 07
Issue: 03 | Mar 2020.