Natural Language Query Processing

Anup S. Kunte[1], Ajinkya Hasbe[2], Abhishek Chavan [3], Kunal Patil[4]

[1]Professor, Department of Information Technology, Konkan Gyanpeeth College of Engineering
[2]B.E. Students, Department of Information Technology, Konkan Gyanpeeth College of Engineering

---***---

Abstract - *A company or organizations data is generally stored in a database. Almost all applications use database to store and retrieve data. Storing and retrieving the data from the database is a difficult task for a naïve user who is unfamiliar with query languages. Natural language query processing thus plays a vital role of converting simple English statement spoken or written into SQL query. Retrieving data from database is not easier for the naïve users and NLP allows fetching it using natural language. Questions in English are fired as a statement. Interactive interface will be provided for user.*

Key Words: Databases, User Interface, Natural Language Processing (NLP), Natural Language Query Processing (NLQP), Structured Query Language (SQL), Natural Language Interface for Database (NLDIB), NL (Natural Language) Java Runtime Environment (JRE), Java Database Connectivity (JDBC), Open Database Connectivity (ODBC)

1. INTRODUCTION

Natural language processing is one of reliable methods for Human-computer interaction. Various methodologies have been developed to implement NLP. NLP is used for machine translation, Information Extraction, Summarization, etc. NLP helps to analyse, understand and derive the meaning from human language. NLP are appropriate in database query systems. Ambiguity among the words is incorporated and an appropriate response is generated. The database is approached and an appropriate response is generated. Many techniques can be used to improve the learning techniques. NLP is characterized as a hard problem in computer science. To understand human language is to understand not only the words, but the concepts and how they're linked together to create meaning. In general, the more data analysed, the more accurate the model will be. Reduce words to their root, or break up text into tokens using Tokenizer. Tokenization is the conversion of an input signal into parts (tokens) so that the computer can process it. For example, even when a machine receives text from the keyboard, the text enters the machine as stream of ASCII characters, which must be classified into individual units, such as words or letter/number characters.

2. PROBLEM DEFINATION

It is need of industries to maintain their data in the digital form. The database provides use the facility to store the data in tabular form which is easier to maintain. The interface to these databases is provided only by the specialized query languages like SQL which are quite complicated and learning it thoroughly takes much time and efforts. People who are unfamiliar to these languages cannot access the data. Due to these anomalies the need to allow the non-administrative user with capability of accessing the database arises.

3. OBJECTIVES

Objectives of our project are as follows:

i. To get English statement from user as input in the form of spoken or written and will convert it into Standard SQL statement.
ii. An interactive interface will be provided for user. The user will use a microphone speaker having disabilities to write or other users can also interact with keyboard.
iii. The converted Standard SQL statement will be fired on database and desired result will be displayed on the screen.
iv. To eliminate the problem of understanding the database query language.

4. BASIC MODULES

A module is a separate unit of software or hardware. Typical components include portability, which allows them to be used in a variety of systems, and interoperability, which allows them to function with components of other systems. Different modules in our database are as follows:

Log-in: A log-in to the system is provided to each and every user who works with that specific database system by an administrator. An administrator to the system can add or delete users.

NLQP system: Actually converts the natural language sentences to SQL query and display the results.

Database: Consist of the records that are to be retrieved by NLQP system.

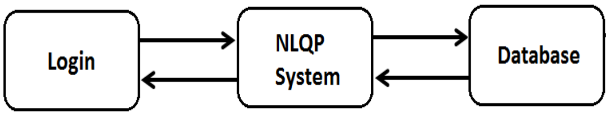


Fig -1: Overview of the System

5. FUNCTIONAL SPECIFICATIONS

i. The user has to first login and then connect to the database.
ii. Then, type the natural language query in the dialogue box given. The query should be based on suggestions or examples given to type a query.
iii. Click on "Execute query" button. Also a "Reset query" button is given if user wants to reset written query.
iv. The system will generate SQL query by dividing each word in the sentence and matching it with its existing word in hash table.
v. Once the SQL query is generated, query will be fired on the database.
vi. The result of the query will be displayed in the Output box.

6. SYSTEM DESIGN

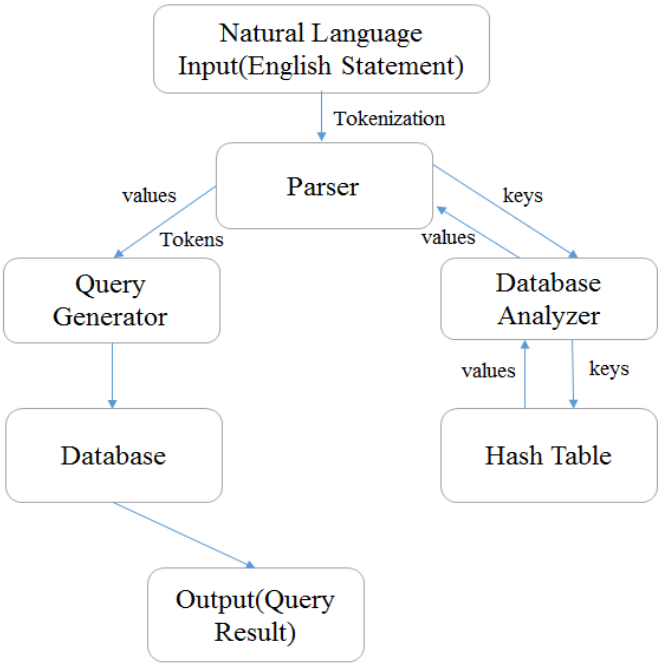


Fig -2: Implementation Logic

To convert an English Statement and retrieve data from database. Every block stated above are important. The functioning of each block is as follows:

NL Input: It is the input statement or sentence in English Language.

Parser: The parser does the job of tokenization of the sentences into words or as tokens.

Database Analyzer: The keys or tokens are then moves to the analyzer to find the matching or related words in the hash table. Ambiguity among the words are handled or managed here.

Hash Table: The values of the keys are identified in the hash table and they are assigned to the keys.

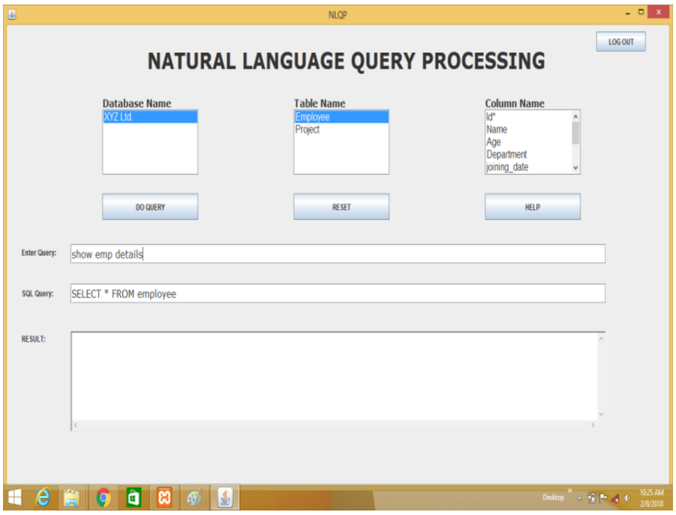
Query Generator: Based on the values equivalent to the keys. A proper SQL query is generated to execute it over the database.

Database: The database consist of data records in the form of rows and columns checks for the desired result.

Output: The output is generated based on the English statement fired at the input.

7. SYSTEM UI

The system UI will consist of a basic login screen for the naïve users. An admin panel for administrator to add or delete users based on the employee durations. A help option for users to guide them to type a proper English statement. The actual conversion screen of English statement to SQL will be as below;



Database Name: XYZ Ltd.

Table Name: Employee, Project

Column Name: Id*, Name, Age, Department, joining_date

Enter Query: show emp details

SQL Query: SELECT * FROM employee

Fig -3: UI for Natural Language to SQL conversion

8. CONCLUSIONS AND FUTURE SCOPE

Natural Language Processing can bring powerful enhancements to virtually any computer program. Retrieving information from the database requires knowledge of technical languages like SQL. In this project we consider a lightweight approach of translating English queries into equivalent SQL queries. In this approach we look at extracting certain keywords and indicators from an English query spoken or written using HashTable technique, and then using a system to generate the query based on the keywords.

Currently our project is built around the fixed database i.e. it can understand the queries about the fixed database only. But the project can be extended to cover any database. Support for other languages should be provided for enhancing the project. The program should identify the vendor of the selected database and should use appropriate drivers to connect them.

ACKNOWLEDGEMENT

We are grateful to our Prof Jitendra P. Patil head of department of Information Technology for making requisite facilities available for us for his special guidance. Our appreciation must be given to our project guide Assistant Prof. Anup S. Kunte who is supporting us throughout project work in the form of comments, corrections and suggestions, which have caught us, several times wondering about the true path. A word of thanks for those entire people who have helped us lot to make for you a successful project. Last but not the least we are thankful to our parents and friends for their constant inspiration, encouragement and well-wishes by which we have made a challenging project.

REFERENCES

[1] IJCSI International Journal of Computer Science Issues, Vol. 8, Issue 2, March 2011 ISSN (Online): 1694-0814

[2] Gauri Rao et al. (IJCSE) International Journal on Computer Science and Engineering Vol. 02, No. 02,2010,219-2

[3] S. Agrawal, S. Chaudhuri, and G. Das. Dbxplorer: A system for keyword-based search over relational databases. In ICDE, pages 5{16, 2002.

[4] D. K¨upper, M. Strobel, and D. R¨osner. Nauda – a cooperative, natural language interface to relational databases. In SIGMOD Conference, pages 529–533, 1993.

[5] Practical Artificial Intelligence Programming in Java Version 1.2, last updated January 1, 2006. By Mark Watson.

[6] A.-M. Popescu, O. Etzioni, and H. A. Kautz. Towards a theory of natural language interfaces to databases. In IUI, pages 149–157, 2003.

[7] A. Simitsis, G. Koutrika, and Y. E. Ioannidis. Pr´ecis: from unstructured keywords as queries to structured databases as answers. VLDB J., 17(1):117–149, 2008.

[8] I. Androutsopoulos, G. D. Ritchie, and P. Thanisch. Natural language interfaces to databases – an introduction. Natural Language Engineering, 1(1):29–81, 1995.

[9] M.-C. de Marne↵e, B. MacCartney, and C. D. Manning. Generating typed dependency parses from phrase structure parses. In LREC, pages 449–454, 2006.

[10] V. Hristidis and Y. Papakonstantinou. Discover: Keyword search in relational databases. In VLDB, pages 670–681, 2002.