Natural Language to Database Interface

Aarti Sawant[1], Pooja Lambate[2] , A. S. Zore
[1] Information Technology, University of Pune, Marathwada Mitra Mandal Institute Of Technology.
Pune, Maharashtra, India

***Abstract*- Information is playing an essential role and one of the foremost sources of information is databases. Almost all IT application are storing and retrieve or extracting information from databases. To retrieve the information one has to be use the Structured Query Language (SQL) for relational database systems. The idea of using Natural Language instead of SQL has prompted the development of new type of processing called Natural language Interface to Database. The natural language interface takes natural language questions as input and gives textual response in the form of data present in the relational database so that informal or non-technical user are not essential to have knowledge of SQL language structures and format hence any user can interact with the databases.**

Keywords: Natural language processing, Question-Answering in NLP, Natural Language to Database Interface, NLDBI Architecture.

1. INTRODUCTION

Information is playing a very important role in our lives today and one of the major sources of information is databases. Databases are gaining major importance in a massive variety of application areas both private and public information systems. Databases are build with the objective of facilitating the behavior of data storage, processing, and retrieval etc. every part of which are coupled with data management in information systems. In relational databases, to retrieve information from a database, one wishes to originate a query in such way that the computer will recognize and fabricate the preferred output. Retrieval of information from the databases needs the knowledge of databases language like the Structured Query Language (SQL).The SQL rule are based on a Boolean interpretation of the queries. But not entire users are known to the Query language and not all user's seeking information from the database can be answered explicitly by the querying system. It is due to the fact that not all the requirement's characteristics can be expressed by regular query languages. To simplify the complexity of SQL and to manipulate of data in databases for common people (not SQL professionals), many researches have turned out to use natural language instead of SQL. The idea of using natural language instead of SQL has led to the development of new type of processing method called Natural Language Interface to Database systems (NLIDB). The NLIDB system is actually a branch of more comprehensive method called Natural Language Processing (NLP). In general, the main objective of NLP research is to create an easy and friendly environment to interact with computers in the sense that computer usage does not require any programming language skills to access the data; only natural language (i.e. English) is required.

2. NATURAL LANGUAGE PROCESSING

Natural Language Processing (NLP) is interdisciplinary by nature. Linguistics and artificial intelligence can be combined to develop computer programs. It helps to coordinate activities of understanding or producing texts in a natural language. Database Natural Language Processing is an important success in NLP.
Asking questions in natural language to get answers from databases is a very convenient and easy method of data access.

2.1 Qa System In Nlp

In QA domain, many QA systems, such as START, Ask have been presented to sustain those users searching the precise information about some topics. With these systems, START may be measured as the finest system that can return the superior answers for users. However, START is only capable to answer questions about concept, but it could not answer the questions concerning causes and methods.

Additionally, we had determined that an open source QA system, that is Open-Ephyra; it is an open framework for question answering. It extracts answers to natural language questions from the complementary sources. It was developed on Java framework.

This system has: a glossary, a set of questions, method to find out the straightforward answers for questions. Once accepting a question, the system organize question into one of definite categories of question to discover and come apart keywords based on the dictionary. Later than, OpenEphyra uses these keywords to search in dataset paragraphs that contain them. The result will be projected adequate degree and the excellent result will be display to user.

Once taking into consideration these QA systems, we presume that recent QA systems perform the question processing based on this theory:

1) Match the query to existing queries form.
2) Generate a set of keywords or a set of queries in knowledge base.
3) Determine the result that fit to the question and generate the answer/result.

In nearly all of the typical NLDBi systems the natural language statement is transformed into an internal representation based on the syntactic and semantic information of the natural language. This representation is then converted into queries using an illustration converter. Prior to a natural language query is translated to a corresponding query in technical language like SQL it has to be moved out through a lot of steps.

1) Preprocessing

2) Grammar Checking

3) Query Generation

4) Data Collection

3. NLDBI ARCHITECTURE

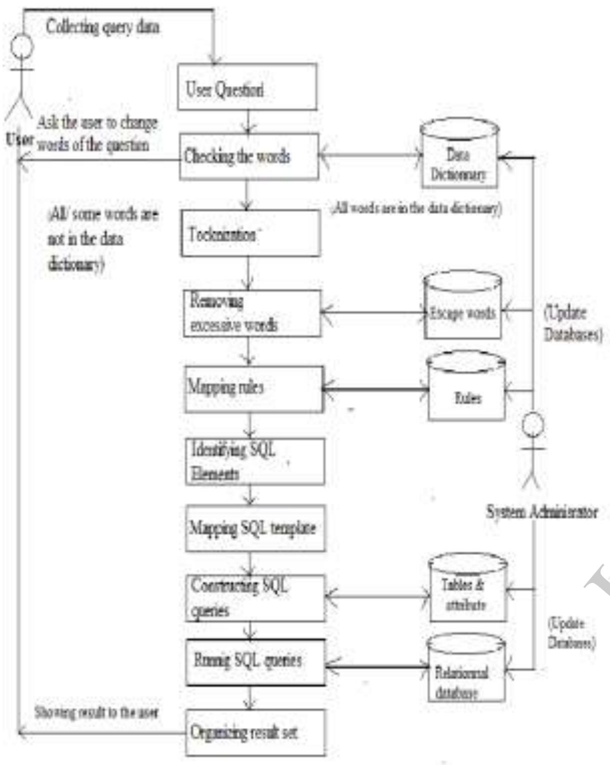

Fig 1: Brief Architecture Of Nldbi System

User question:
Input from the user is given in the form of questions (Like what, who, where).All the values in the input natural language statement have to be in double quotes which yield to identify the values from the user input statement.
The user will enter his question in natural language. E.g. English is considering as most preferable natural language. The question to be asked is decided by user by collecting the query data; this question is input to the system on which further processing will carried out.

Checking words:
The question in string words of asked by user are checked against the words present in the data dictionary. The question contains string of words which should be present in the data dictionary. If word in the question does not match with data dictionary the user has to change the words in the question.

Tokenization:
This module splits the string of question into number of tokens. Each token is identifying by unique identifier and order number is assign to each token. It is very important to tokenize a question so that the text can undergo further processing.

Removing excessive words:
This module removes the excessive words from the question which does not play important role in the application. A, an, are, the, is, was, were etc. these are the escape words need to be removed.

Mapping rules:
After removing excessive words some rules have to be mapped and roles are defined by the system administrator who works on system database. Rules are mapped with user input question statement.

SQL element identifier:
This module identifies each and every SQL element present in the input question entered by user. Elements are identified using rules and SQL template string.

Mapping SQL template:
All SQL elements are stored in particular array. SQL template is constructing using available SQL elements.

Constructing SQL query:
SQL query is formed by constructing SQL template and differentiating SQL elements.

Algorithm to generate SQL query from SQL template

```
Begin
       Store the SQL template string to the
       variable U

       If (the value of variable U equals
       ‘tocken1’) then
       Store the name of the identified
       tables in array V
Do
       Store the value of variable V to T
       Get the default attribute for the
       table name T and store it to the
       variable A
       Construct a SQL query as
       “SELECT DISTINCT A FROM T” and store
       it to variable W

       While for each table in array V

End If
End
```

Data Collection:
This module collects the output of the SQL statement and displayed it in the user interface as a outcome of SQL query.

3.1 Algorithm used in NLWIDB System
The following stages illustrate the algorithm for the NLDBI system:

1 . Check for a user question value and if value is exist Then
Remove the value from the question string.
Check for all words in question string which occurs in data dictionary.

2. Tokenization (scanning)

- ➢ Fragment the question string into the tokens.
- ➢ Assign index number to each token identified.

3. Removing extreme words in question sequence or string.

4. Mapping rules by eliminating preceding words from question string.
- ➢ If matches with a rule, remove the matched string from the question string and make the remaining as the question string.
- ➢ Map the rule till you have question string becomes null.

5. Store the identified SQL elements in arrays with its index.

6. Build an SQL pattern string by considering available number of SQL elements.

3.2 Following example shows how the query is formulated from the natural language statement:

User will enter input in natural language statement:
What is the salary of employee where employee Id = "101"?

Checking the words:
What is the salary of employee where employee Id = "101"?
All words are present in data dictionary.

Tokenization:
Token [0]: What
Token [1]: is
Token [2]: the
Token [3]: salary
Token [4]: of
Token [5]: employee
Token [6]: where
Token [7]: employee
Token [8]: id
Token [9]: =
Token [10:] "101"

Removing excessive words:
The question string after removing excessive words:
Salary of employee where employee id =”101”

Mapping rule:
Salary of employee is mapped with: emp_salary table
Employee id is mapped with: emp_ID

Identifying SQL elements:
Attribute: emp_salary
Attribute: emp_ID
Value: 101

Mapping SQL template:

The string code for finding SQL template [emp_salary, emp_ID]
Found SQL template.

Constructing SQL query:
Select emp_salary from employee where emp_ID =”101”;

Running SQL query result will be:

Emp_salary
25000

3.3 ADVANTAGES

1. *No use of artificial language*
In this system the users input is entirely in natural language which is most easy for interaction with the database this principle makes the system more flexible and scalable.

2. *Fault tolerance*
Most of NLDBI systems offer some tolerances to negligible grammatical errors, while in a computer system; most of the time, the lexicon should be precisely the same as defined, the syntax should properly follow definite rules, and any errors will cause the input automatically be rejected by the system.

3. Easy access for Multiple Database Tables
Queries that include multiple database tables like "Show the list of employee where employee salary is greater than 20,000 and dept=’marketing’ " such queries are also provides the accurate results.

3.4 Limitations

NLDBI system are domain specific:
Almost all NLDBI systems are domain specific so that it is difficult to interface with the other databases, in such case it is necessary to regularly update database dictionaries consistently.

4. CONCLUSIONS

Natural Language Processing can bring dominant enhancements to almost any database interface. The Natural Language to Database Interface (NLBDI) is no exclusion because we presented the NLDBI system which converts an extensive range of text queries (English questions) into prescribed ones that can then be executed against a database by employing strong language processing techniques and methods. This study illustrates that NLDBI provides a convenient as well as consistent means of querying access, hence, a genuine potential for associating the gap between computer and the normal end users.

ACKNOWLEDGMENTS

This is a small review of my B.E. thesis work that we are going to implement. I specially thank to my guide for his assistance and guidance.

REFERENCES

1. Natural language Interface for Database: A Brief review IJCSI International Journal of Computer Science Issues, Vol. 8, Issue 2, March 2011www.IJCSI.org
2. Natural Language Interface for Database Proceedings of the Third International Symposium, SEUSL: 6-7 July 2013, Olivia, Sri Lanka
3. Incremental Information Extraction Using Relational Databases[IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, VOL. 24, NO. 1, JANUARY 2012] Luis Tari, Student Member, IEEE Computer Society, Phan Huy Tu,Jo¨ rg Hakenberg, Yi Chen, Member, IEEE Computer Society,Tran Cao Son, Graciela Gonzalez, and Chitta Baral
4. GenerIE: Information Extraction Using Database Queries.Luis Tari1, Phan Huy Tu1, Jorge Hakenberg1
   Department of Computer Science and Engineering, Arizona State UniversityTempe, AZ 85287, USA
5. Efficient Information Extraction over Evolving Text Data Jun Yang, Raghu Ramakrishnan Duke University, Yahoo! Research

Biographies: Prof. A. S. Zore is currently working as a Lecturer in Marathwada Mitra Mandal Institute of Technology, Pune, India. His research interests are Data Mining, etc.