

A SIMPLIFIED MODEL OF NATURAL LANGUAGE INTERFACE FOR QUERYING DATABASE

Ashwath .M¹, Balapradeep K.N², Savitha C.K³

^{1,2,3}Department of Computer Science, KVG College of Engineering.

Abstract: *Natural Language Interfaces to Database is a type of database interface that allows the user to access the data using natural language. Using query language for dealing with databases is always a professional and complex problem. This intricacy causes the client's use of information existing in database points of confinement to utilize clear reports there are in some pre executed programming projects. However, user can create this opportunity that each none professional user transfers his questions and requirements to computer in natural language and derives his desired data by natural language processing. This paper represent a method for building a "Natural Languages Interfaces with Data Bases" (NLIDB) system. This system prepares an "expert system" implemented in prolog which it can identify synonymous words in any language. It initially parses the input sentences, and then the natural language expressions are transformed to SQL language.*

Index Terms— *Natural Language Processing, Database Management Systems, Structured Query Language.*

I. INTRODUCTION

In the current scenario of fast computing, computer-based information retrieval technologies are being widely used to help academic organizations and educational institutions, companies to manage their systems and information processes. These are used to manage data that are capable of managing different types of data that are stored in databases also known as DBMS (Rukshan et al., 2013). Regardless of the retrieval of information from a large amount of information that is competent in the social databases, the client still needs the DB dialect / construction to plan the queries completely. Artificial Intelligence (AI) and Linguistics can be combined to develop programs that can help understand and produce information in a natural language. NLP based on databases is, therefore, an important success in the processing of natural language. It is a convenient way to access data by asking questions in natural language to get answers, since a layman may not understand the query language of the database. An NLQ interface for the database system is an application that accepts a query in natural language, creates an SQL query from there and executes it to retrieve the data from the relational database. The result obtained from the database is a sequence of elements. The query is

generated by identifying the lexical relations of the elements of the NLQ (Gupta and Sangal, 2012). The construction of robust and applicable NLIDBS has become more acute in recent years. The amount of information on the Internet has grown steadily and a wider population can now access data stored in a variety of repositories through web browsers (Nihalani et al., 2011). Therefore, NLIDBS are designed to optimize search results and produce information more accurately. The present investigation extends the existing work by processing more complex queries together with the elimination of ambiguities [1].

A Simplified Model Of Natural Language Interface For Querying Database is a long-standing open problem and has been attracting considerable interest recently. The application which converts the natural language question to MySQL query and executes it in the android application. The user no needs to know about the syntax of MySQL. There two applications one will run on server and other on User mobile. User application will translate the natural language to English (US) using Google API and it send it to Server application will convert it to MySQL query and sends to user as response which will execute in user app to obtain result.

II. RELATED WORK

Different technique are used to develop NLIDB systems such as

1. Pattern matching system
2. Syntax based system
3. Semantic based system

A. Natural language processing for speech synthesis

A system and the method interact with networked objects, via a computer using the utterance, speech processing and natural language processing. A data definition file relates networked objects and a speech processor. The speech processor searches a first grammar file for a matching

phrase for the utterance, and search a second grammar file for a matching state if the coordinating expression isn't found in the principal sentence structure record. The system also include a natural language processor to search a database for a matching entry for the matching phrase. The natural language processing is the computerized approach to analyzing text and being a very active area of research and development. This is based on the text to speech conversion in which the text data is first input to the system. It uses high levels of modules for speech synthesis. It uses sentence segmentation which deals with punctuation mark with simple decision tree [2].

B. Natural language processing for speech recognition

Automatic speech recognition system makes use of natural language processing system based on grammars. It uses context free grammar for representing syntax of that language presents a means of dealing with spontaneous through the spot lighting addition of automatic summarization including indexing, which extracts the gist of the speech transcription in order to deal with the information retrieval and dialogue system issue[3].

C. Lunar

Prototype NLIDB had already appeared in the late sixties and seventies. The best known NLIDB of the period is LUNAR. The LUNAR was the first system that was informally introduced which answered question about the samples of the rock. The performance of the lunar was very impressive and easily handled. Another system called LADDER was proposed. Which was designed for US navy shifts used synchronic linguistic it was implemented entirely in prolog. it was used to transform English languages into prolong expressions, which were evaluated on the prolong database. This was very impressive, efficient and sophisticated kind of system[4].

D. A bit of progress in language modeling

In the previous quite a while, various distinctive dialect displaying upgrades over basic trigram models have been discovered, including getting, higher-arrange n-grams, skipping, inserted Kneser-Ney smoothing and bunching. This paper present explorations of variations on or of the limits of each of these techniques including showing that sentence mixture models may have more potential. While all of these techniques have studied separately, They have been rarely been studied separately, they have rarely been studied in combinations. We compare a combination of all techniques to a Katz smoothed trigram model with no count cutoffs . We accomplish perplexity lessening in the

vicinity of 38 and half relying upon preparing information estimate, and in addition a word blunder rate decrease of 8.9%. Our perplexity diminishment are maybe the most noteworthy detailed contrasted with a baseline[5].

III. PROBLEM DEFINITION

The project aims to understand any natural language input statement and convert it to an query. It involves understanding all natural languages that can be inputted by the user and finding and solving complex relationships between words and inputs given in the statement. Natural language processing interface with database also aims to resolve problems in the existing systems that try to understand natural languages. The problems such as use of complex interfaces that might confuse users, various syntaxes or predefined statements that need to be followed, Limited application and scope, Lack of modifiability and modularity. While easing the process of programming for the user, Natural language processing interface with database also tries to ease the effort required by the programmers to build the library for Natural language processing interface with database. Hence separate user interfaces to help developers modify/enhance the libraries for NLIDB. This NLIDB tries to reduce the amount of information to be entered by the programmer to add new features to it.

A. Scope of the project

More than a project, Natural language processing interface with database is a concept/method to understand natural language. Natural language processing interface with database will prove to be helpful for different types of user. Common people with no programming knowledge will benefit from it by being able to instruct computers to perform complex task using simple statements. By implementing Natural language processing interface with database as a digital assistant in mobile devices, users will be able to make complete use of the hardware capability of the devise, which otherwise is limited by the number of apps its user has installed. Programmers will be able to get more productive by being able to generate error free codes quickly without the need of writing long instructions. This will also improve the development process by greatly reducing development and verification time. NLIDB can also prove to be a great tutor for budding programmers. It will help in their step by step evolution into good programmers. The easy and hustle free interface will prove to be of great advantage in such scenarios. NLIDB would eliminate the necessity for students to mug up syntaxes and rules of coding and help them concentrate more on creativity and innovation.

IV. IMPLEMENTATION

The project is developed using the python and In2sql tool this is only a prototype which can be implemented further. This project is implemented has two application software one is client side and another one is server side. Client side application is developed using android studio. Java, XML, Sqlite programming language is used. Server side application is developed using python programming language.the database sqlite is used for storing student data. Following are the Software used for proposed system for getting good subset of feature from given data sets.

A. Android technology

Android is an open source programming stack that joins the working structure, middleware, and key applications nearby a game plan of API libraries for creating adaptable applications that can shape the look, feel, and limit of cutting edge cell phones. Android was produced from the earliest starting point enable designers to make persuading adaptable applications that take full good position of each one of the a handset conveys to the table. It is attempted to be really open. For example, an application could unending supply of the phone's middle helpfulness, for instance, making calls, sending texts, or using the camera, empowering architects to make wealthier and more sturdy experiences for clients.This is valid, as an engineer you can do everything, from sending short messages with only 2 lines of code, up to supplanting even the home-screen of your gadget. One could without much of a stretch make a completely tweaked working framework inside weeks, giving no a greater amount of Google's default application to the client. Android is based on the open Linux Kernel. Moreover, it uses a custom virtual machine that has been intended to upgrade memory and equipment assets in a versatile domain. Android will be open source; it can be generously reached out to join new bleeding edge advancements as they develop. The stage will keep on evolving as the engineer group cooperates to assemble inventive versatile applications.

B. Java

Java is a programming dialect. It was initially created by James Gosling at Sun Microsystems, which is presently a piece of Oracle Corporation. It was discharged in 1995 as a piece of Sun Microsystems' Java stage. The dialect has grown quite a bit of its linguistic structure from C and C++. Java applications are generally gathered to byte code (class document) that can keep running on any Java Virtual Machine. Java is as of now a standout amongst the most mainstream programming dialects being utilized. It has

around 10 million clients. Java and JavaScript are not the same thing; truth be told, they are not by any means related. JavaScript is a scripting dialect which is utilized chiefly within web programs. Java is object oriented. Unaltered C++ or C code won't work in Java, as a rule, however Java looks much like C and C++. Java can keep running on a wide range of working frameworks. This makes Java stage autonomous. Java does this by making the Java compiler transform code into Java byte code rather than machine code. This implies when the system is executed, the Java Virtual Machine deciphers the byte code and makes an interpretation of it into machine code.

C. Python

Python is an open source programming dialect. Python was made to be anything but difficult to peruse and capable. A Dutch developer named Guidovan Rossum made Python in 1991. He named it after the Monty Python's Flying Circus network show. Python is a translated dialect. Translated dialects do not should be aggregated to run. A program called a mediator executes Python code on a PC.. This means that a programmer can change the code and quickly see the results. This also means that Python is slower than a language compiled as C, since it is not running machine code directly.

D. In2sql

In2sql is an NLP tool for consulting a database in natural language. The tool takes in input a database model and a sentence and translates the latter into a valid SQL statement capable of querying the input data model. The tool can handle any language, as long as it has its configuration file (that is, a file with the language's keywords). The language configuration files can be found in lang / directory. The files are CSV files. Each line represents a type of keywords. Anything before the colon is ignored. Keywords must be separated by a comma

Checking for non-English

Here the input language is checked for non English by pattern matching technique the pattern used is English alphabets A-Z. If input string is non-English the string is passed to Google Translation API for translating the string to English.

```
Algorithm
Pattern="^[A-Z a-z 0-9]+$"
If(input matches(pattern)
Then English
Else
Non English
```

Google Translation API

The non English string is passed to the Google Cloud Translation API. The Google Cloud Translation API can dynamically translate text between thousands of language pairs. The Cloud Translation API lets websites and programs integrate with the translation service programmatically. The Google Translation API is part of the larger Cloud Machine Learning API family.

Sending English natural language query to server

The query is send to server by using HTTP GET request the string is passed along with the URL.

Receiving sql query from server

The query received from the server by HTTP response. Conversion of natural language to mysql query has following steps.

- Tokenization
- Part of speech tagging
- Classifying attributes and clauses
- Generation of sql query

V. RESULTS

A snapshot is defined as a photograph that is "shot" spontaneously and quickly, the state system at a particular point in time. The snapshot shows the user interface at the NLIQDB, where the user observe two text boxes and two buttons for add record and getquery. First the user has to add record to the system by entering the student details and then enter the getquery button.

After successful adding of records the user can enter the query in natural language to fetch the information from the database which will be converted to sql queries.

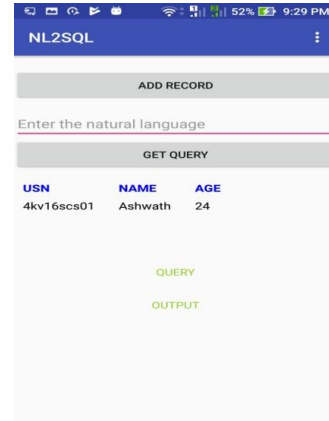


Fig 1: Main Screen

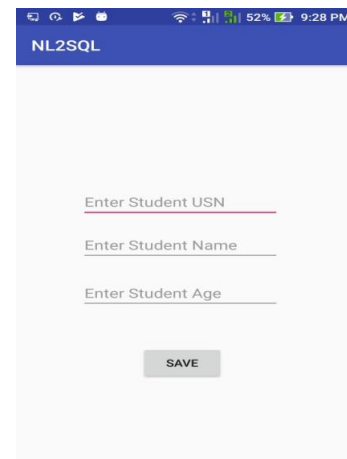


Fig 2: User interface for entering student details

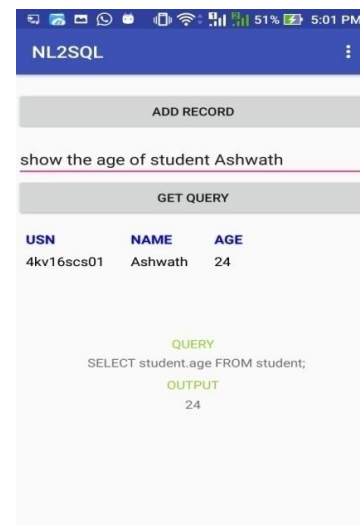


Fig3 : Fetching Information

VI. CONCLUSION

This is a basic version; hence its functionalities are limited. There is little restriction in giving input in the natural language, such as user should maintain some sequence in giving input i.e. user is not independent of giving input on his wish. But, the successful completion of this prototype proves the possibility of efficiently understanding complex instructions provided by the user using this approach.

REFERENCES

- [1]. Garima Singh, Arun Solanki; "An algorithm to transform natural language into SQL queries for relational databases" Department of Computer Science and Engineering, Gautam Buddha University, Greater Noida, India, September 2016
- [2]. D.G. Bobrow, R.M. Kaplan, M. Kay, D.A. Norman, H. Thompson, and R. Winograd, "GUS, Frame-Driven Dialog System," *Artificial Intelligence*, vol. 8, no. 2, pp. 155-173, 1977.
- [3]. Jyothsna Cherapananjeri, Lavanya Lingareddy Himabindu K "Keyword based question answering system in natural language interface to database", *International Journal of Advanced Research*. In *Computer Engineering and Technology(IJARCET)*, vol.3 issue 12, December 2014.
- [4]. D.M. Dilts and W. Wu, "Knowledge-Based Systems for Integrating Computer Integrated Manufacturing Databases," *Proc. Third Int'l Conf. Expert Systems Leading Edge Production Operations Management*, pp. 255-268, 1989.
- [5]. Alpa Reshamwala, "Review on natural language processor", *An International Journal (ESTIJ)* ISSN:2250-3498 vol.3,NO.1, February 2013.