International Journal of Computer Applications (0975 – 8887)
International Conference on Microelectronics, Circuits and Systems (MICRO-2014)

Automatic SQL Query Formation from Natural Language Query

Prasun Kanti Ghosh
Student,IEM Kolkata
SaltLake,Kolkata

Saparja Dey
Student,IEM Kolkata
SaltLake,Kolkata

Subhabrata Sengupta
Assistant Professor,IEM Kolkata
SaltLake,Kolkata

ABSTRACT
Natural language processing is a field of computer science concerned with the interactions between computers and human (natural) languages. It is becoming one of the most active areas in the interaction between human and computer. These include spoken language systems that integrate speech and natural language. It is an interdisciplinary research area at the border between linguistics and artificial intelligence, aiming at developing computer programs capable of human-like activities like understanding or producing texts or speech in a natural language, such as English or conversion of natural language in text or speech form to languages like SQL. The most important applications of natural language processing include information retrieval and information organization, machine translation. The goal of NLP is to enable communication between people and computers without resorting to memorization of complex commands and procedures.

General Terms
Natural Language Processing; SQL Query; SL4A; Android; Python.

Keywords
Natural language query; Speech-to-text; log file; data dictionary; speech recognition

1. INTRODUCTION
The main purpose of Natural Language Query Processing is for an English sentence to be interpreted by the computer and appropriate action taken. Despite the challenges, natural language processing is widely regarded as a promising and critically important endeavor in the field of computer research. The applications that will be possible when NLP capabilities are fully realized are impressive computers would be able to process natural language, translating languages accurately and in real time, or extracting and summarizing information from a variety of data sources, depending on the users' requests.

Asking questions to databases in natural language is a very convenient and easy method of data access, especially for casual users who do not understand complicated database query languages such as SQL. This system focuses on the solution of the problems arising in the analysis or generation of Natural language text or speech, such as syntactic and semantic analysis or compilation of dictionaries and grammars necessary for such analysis. It proposes the architecture for translating English Query into SQL using Semantic Grammar.

2. RELATED WORK DONE
ELIZA (Joseph Weizenbaum, 1964) is a simulation of a Rogerian psychotherapist, written by Joseph Weizenbaum between 1964 to 1966. Using almost no information about human thought or emotion, ELIZA sometimes provided a startlingly human-like interaction. When the "patient" exceeded the very small knowledge base, ELIZA might provide a generic response. For example, responding to "My head hurts" with "Why do you say your head hurts?"[12]

SHRDLU (Terry Winograd, 1968) was an early natural language understanding computer program, developed by Terry Winograd at MIT in 1968–1970. With it, the user carries on a conversation with the computer, moving objects, naming collections and querying the state of a simplified "blocks world", essentially a virtual box filled with different blocks.It was written in the Micro Planner and Lisp programming language on the DEC PDP-6 computer and a DEC graphics terminal. Later additions were made in the computer graphics labs at the University of Utah, adding a full 3D rendering of SHRDLU's "world".[13]

LUNAR (Woods, 1973) involved a system that answered questions about rock samples brought back from the moon. Two databases were used, the chemical analyses and the literature references. The program used an Augmented Transition Network (ATN) parser and Woods' Procedural Semantics. The system was informally demonstrated at the Second Annual Lunar Science Conference in 1971. [1]

LIFER/LADDER (Hendrix, 1978) was one of the first good database NLP systems. It was designed as a natural language interface to a database of information about US Navy ships. This system, as described in a paper by Hendrix, used a semantic grammar to parse questions and query a distributed database. The LIFER/LADDER system could only support simple one-table queries or multiple table queries with easy join conditions. [6]

QUESTION-ANSWERING SYSTEM (Nguyen Tuan Dang, 2009) proposed a method to build a specific Question-Answering system which is integrated with a search system for e-Books in the library. Users can use simple English questions for searching the library with information about the needed e-Books, such as title, author, language, category, publisher… In this research project, the main focus is on fundamental problems in the natural language query processing: approaches of syntax analysis and syntax representation, semantic representation, transformation rules for syntax structure of semantic structure. [4]

3. PROBLEM DESCRIPTION
A huge amount of labor is required if we wish to obtain only the required information from the entire repository of information system.Natural language processing is a process by which the user query (entered in English language) in natural language will be converted to a SQL query based on the query entered.

Any ordinary person is not expected to know the SQL language, and hence this system would help him in generating the same, so that information retrieval is easier for the database, as database understand the SQL language only.

The objective is to parse the query and with the help of the dictionary, carry out different phases like morphological analysis, syntactic analysis, semantic analysis etc. and finally generate the SQL query.

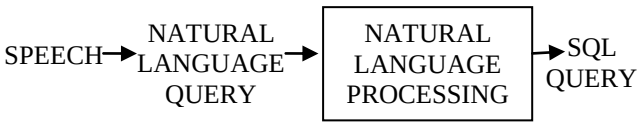

Figure 1: Problem Description

Consider a database, say DB. Within this DB database we have placed certain tables with attributes, which are properly normalized. Now if the user wishes to access the data from the table, he/she has to be technically proficient in the SQL language to make a query to the DB database. Our system eliminates this part and enables the end user to access the tables in his/her language.

Let us take an example:

Suppose if we want to view information about a particular student from the Student table, then we are supposed to use the following query:

SELECT * FROM employee WHERE clg_name='<college name >';

But a person, who doesn't know SQL, will not be able to access the database unless he/she knows the syntax and semantics of firing a query to the database. But using NLP, this task of accessing the database will be much simpler. So the above query will be rewritten using NLP as:

Give the information of the employee who works in the college <college name>. Both the SQL statement and NLP statement to access the Student table would result in the same output the only difference being, a normal person who doesn't know anything about SQL can easily access the DB database.

4. SYSTEM DESIGN

We can explain what is the actual process carried out within the Natural Language Processing system by means of a method which is also known as "Levels of Language" also known as Synchronic Model of language. The previous sequential model hypothesis is based on the fact that the levels of Human Language Processing system follow one another in a strict and sequential manner. According to psycholinguistic research, the levels within a language processing interacts in various orders and hence it is much more dynamic.

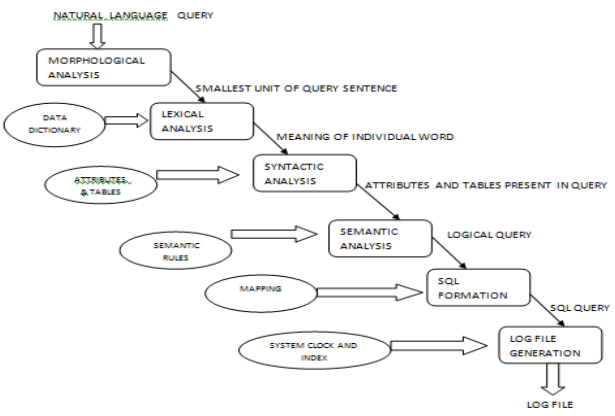

Figure 2: System Design

4.1 Morphology

In this phase, the sentence is broken down into tokens-the smallest unit of meaning. At this level, we split the given input query sentence in natural language into all the words it contains and store the words in a list. For example, if the given input query is "find salary of the employee", then in this phase, each word of the sentence, i.e. find, salary, of, the, employee will be stored in a list like ['find', 'salary', 'of', 'the', 'employee']

4.2 Lexical

At this level, humans, as well as NLP systems, interpret the meaning of individual words. Each word of the tokenized sentence will be mapped with the meaning of the same word present in the data dictionary. For example, from the list generated in the morphology phase, the words will be mapped as "find: select", "salary: salary", "employee: employee".

4.3 Syntactic

At this level at first we find the attributes present in the given input query from the words generated in the lexical phase. Each of them is checked with the attributes in the dictionary which contains all the tables along with their attributes. And then we find the tables which contain the attributes of the given input query. For example, of the output generated in the previous phase, we derive the attributes in the query as "salary" and which belongs to table "employee"

4.4 Semantic

Semantics focuses on the study of meaning of the words present in the natural language query and the relation between signifiers like words, signs, phrases and what do they actually stand for. A field of semantics called Linguistic semantics deals with the study of meaning which interprets human expression through language. This level deals with checking the different conditions like where clause, relational operators, aggregate functions, natural join and build the SQL query accordingly. The final SQL query after checking all the conditions is "select salary from employee"

4.5 Speech to Text Recognition: Python for Android

The Python for android project includes a Python module called "android" and with the help of this module the various methods of the android module can be used within the Python code. It consists of multiple parts which are mostly there to facilitate the use of the Python API. This module is not designed to be comprehensive.

SL4A: The Scripting Layer for Android (abridged as SL4A, which was previously named Android Scripting Environment or ASE) is a library that allows the creation and the running of scripts written in various scripting languages directly on Android devices. SL4A is specially designed for developers and is still alpha quality software.

These scripts have access to many of the APIs available to normal Java Android applications, but with a simplified interface. Scripts can be run interactively in a terminal, or in the background using the Android services architecture. Currently supported languages are:

- Python using CPython
- Perl
- Ruby using JRuby
- Lua
- BeanShell
- JavaScript
- Tcl

With the help of SL4A the python code can be run within the Android system.

Speech Recognition Process in Android:

The main portion of Speech to text Android API is the package android.speech and a class within it called android.speech.RecognizerIntent. Hereby an Intent of the name android.speech.RecognizerIntent gets activated which shows a dialog box whereby it can recognize the input in speech. With the help of this Activity, the input speech is converted to text and this result is sent back to the calling Activity.

The Android module needs to be imported within the Python source code at first. An object for the Android module is created which further calls its various methods to perform the speech recognition function.

This class provides access to the speech recognition service. This service allows access to the speech recognizer. This class cannot be instantiated directly, instead, call createSpeechRecognizer (Context) from the main method. This class's methods must be invoked only from the main application thread.

The implementation of this API is likely to stream audio to remote servers to perform speech recognition. As such this API is not intended to be used for continuous recognition, which would consume a significant amount of battery and bandwidth.

SQL in Android:

SQL is embedded into every Android device. Using an SQL database in Android does not require a setup procedure or administration of the database.

We only have to define the SQL statements for creating and updating the database. Afterwards the database is automatically managed by the Android platform.

5. COMPONENTS OF THE SYSTEM

- Graphical User Interface: The front end or the user interface where the user can choose whether he/she will input the query in Natural Language in text form or speech form.
- Data dictionary: Stores different words and how they can be related to different words belonging to a SQL query. Also stores all the tables along with their attributes.
- Speech-to-text: The user can speak and enter the query, without having to type it.
- Sl4A: The Scripting Layer for Android is a library that allows the creation and running of scripts written in various scripting languages directly on Android devices
- Parser: Derives the Semantics of the Natural Query given by the user and parses it in its technical form.
- Attribute Generation: Examines all the tables stored and finds out all the attributes of all the tables. Then it finds all the attributes present in the natural language query provided by the user.
- Table Generation: Examines all the attributes present within the query provided by the user and finds the tables to which the attributes belong to.
- Query Generation: After the successful categorization of the keywords the user, the system generates a query against the user statement in SQL.
- Log file Generation: The final query along with an index number and time of the query using the system clock, so that we can retrieve the query generated at a particular time giving the time as input.

6. ALGORITHM

- Accept the input from the user either in the form of speech and convert it to text or directly in the form of text.
- If you take the input in speech, then convert it to text by Speech Recognition using Android.
- Split the input query and store it in a list, i.e. tokenize the input sentence.
- Find all the attributes of all the tables.
- Examine the query and find the table present in the query and the attributes present in the query.
- Find the attributes which belong to table present in the query.
- Find the attribute which do not belong to the table in the query (if any).
- Now find the tables which will contain the pair of ((attribute which do not belong to the table in the query), (other attributes present in the table in the query)).
- Select any one table. Thus we will obtain the tables required for natural join.
- For a natural join query, find out the common attribute of the 2 tables and form the inner query. Then form the outer query according to the different conditions.. Merge both of them and generate the final query.
- For a simple query, generate the final query by checking the different conditions accordingly.
- If there are 2 tables, then perform a natural join on the 2 tables with appropriate attributes of the tables.
- Obtain the conditions of the where clause (single condition or multiple condition by finding the “and” word in the input query), aggregate function (checking whether any aggregate function (like sum, avg, count , etc) present in the query) and the relational operators between the conditions from the list of attributes. Add these to the final query.
- Print the final query.
- Write the generated SQL query in a “log file” along with an index number and time of the query using the system clock, so that we can retrieve the query generated at a particular time giving the time as input.

7. RESULTS

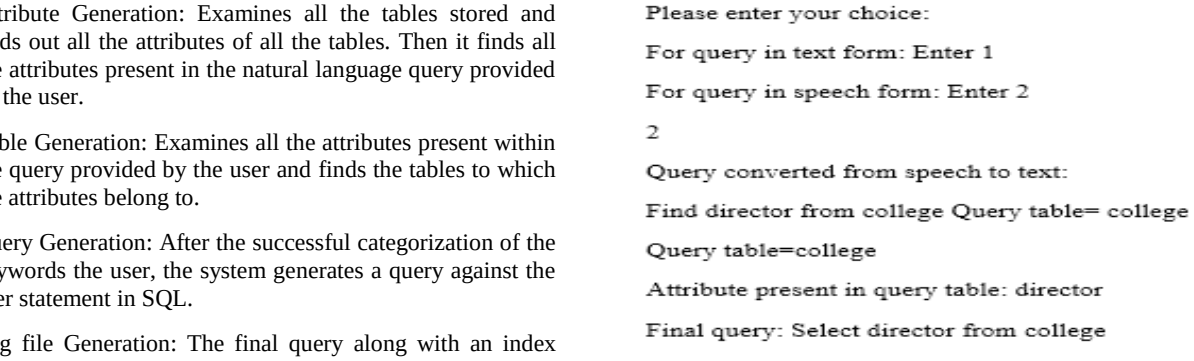

Figure 3: Output of system for input in speech

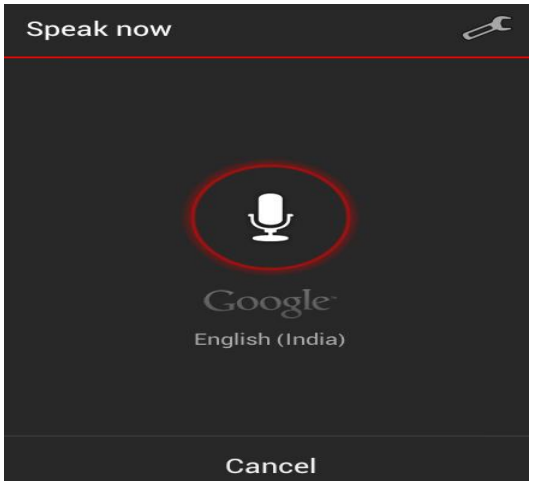

Figure 4: Interface for taking input in speech

Figure 5: Interface for taking input in speech

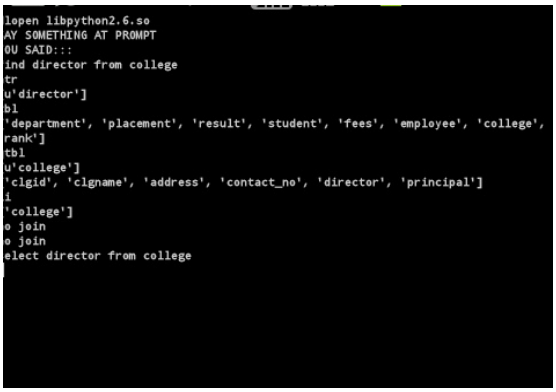

Figure 6: Output of system in console

8. FUTURE IMPROVEMENTS

- The word set that can be recognized by the speech to text recognizer can be expanded.
- Some more complex query handling can be added to the system.
- Standard dictionaries like the Oxford dictionary can be added to the system to enhance its efficiency.
- Currently we are building this system for a single user. It can be expanded for many users and entries for each user can be recorded within the log file along with a user ID.

Please enter your choice:
For query in text form: Enter 1
For query in speech form: Enter 2
1
Enter the query:
Find director of college whose tuition_fees is 20000

Query table=college
Attributes present in query table: director
Attributes not present in query table: tuition_fees
Required table for natural join=["fees","college"]
Common attribute of two tables= college_id

Inner query: (Select college_id from fees where tuition_fees ="20000")

Final query: Select director from college where college_id=(Select college_id from fees where tuition_fees ="20000")

Figure 7: Output of system for input in text

INDEX TIME QUERY

1 Sat Mar 15 12:38:15 2014 select fname, lname from student
2 Sat Mar 15 12:38:24 2014 desc student
3 Sat Mar 15 12:38:24 2014 select avg(salary) from employee
4 Sat Mar 15 12:38:24 2014 select salary from employee where fname="X"
5 Sat Mar 15 12:38:24 2014 select director from college

Figure 8: Log file

English Query: "Find director from college whose tuition_fees is 20000"
Meaning of Query: "Director of that college where tuition fees is 20000"
SQL Query: "Select director from college where college_id= (Select college_id from fees where tution_fees = '20000')"

- The log file can contain more details.
- The database for which we have built the system can be expanded with more attributes as well as tables.

9. CONCLUSION

Natural Language Processing is a very powerful tool which can change the complete working of the computer program interface. This system is currently capable of handling simple queries along with some complex queries. Because not all forms of SQL queries are supported, further development would be required.

10. ACKNOWLEDGEMENT

The satisfaction that accompanies the successful completion of any task would be incomplete without the mention of people whose ceaseless cooperation made it possible, whose constant guidance and encouragement crowned all efforts with success.

We are grateful to our project guide Mr. Sourav Saha for the guidance, inspiration and constructive suggestion that helped us in the preparation of this project.

We would like to thank Institute of Engineering and Management for allowing us to do this project successfully.

11. REFERENCES

[1] Huangi,Guiang Zangi, Phillip C-Y Sheu "A Natural Language database Interface based on probabilistic context free grammar", IEEE International workshop on Semantic Computing and Systems 2008

[2] A.M. Riad, El-Minir, H.K., ElSoud, M.A., Sabbeh. PSSE: An Architecture for a Personalized Semantic Search Engine. International Journal on Advances in Information Sciences and Service Sciences Volume 2, Number 1, March 2010, pp.102 - 112J.

[3] Myungjin Lee, Wooju Kim, Sangun Park. Semantic Association-Based Search and Visualization Method on the Semantic Web Portal. International Journal of Computer Networks & Communications (IJCNC), Vol.2, No.1, January 2010, pp.140 – 152.

[4] Nguyen Tuan Dang, and Do Thi Thanh Tuyen. Natural Language Question Answering Model Applied To Document Retrieval System. World Academy of Science, Engineering and Technology 51 2009, pp.36 – 39

[5] William B. Cavnar, John M.Trenkle. 1994 N-Gram-Based Text Categorization. Proceedings of SDAIR-94, 3rd Annual Symposium on Document Analysis and Information Retrieval, pp. 161—175

[6] Hendrix, G.G., Sacerdoti, E.D., Sagalowicz, D., Slocum, J. "Developing a natural language interface to complex data", in ACM Transactions on database systems,*1978*,pp.105-147.

[7] A.Jebaraj Ratnakumar 2005 An Implementation of Web Personalization Using Web Mining Techniques. Journal of Theoretical and Applied Information Technology, 2005, pp.68–73

[8] Gauri Rao, Chanchal Agarwal, Snehal Chaudhry, Nikita Kulkarni, Dr. S.H. Patil (IJCSE) International Journal on Computer Science and Engineering Vol. 02, No. 02, 2010, 219-223

[9] Gauri Rao, Chanchal Agarwal, Snehal Chaudhry, Nikita Kulkarni, Dr. S.H. Patil (IJCSE) International Journal on Computer Science and Engineering Vol. 02, No. 02, 2010, 219-223

[10] Enikuomehin A.O, Okwufulueze D.O ,An Algorithm for Solving Natural Language Query Execution Problems on Relational Databases by in (IJACSA)International Journal of Advanced Computer Sciemce and Applications, Vol. 3, no. 10, 2012

[11] Sachin Kumar,Ashish Kumar, Dr. Pinaki Mitra, Girish Sundaram (ERCICA) Emerging Research in Computing, Information, Communication and Applications pp:291-298, in proceedings of International Conference., 2013.

[12] Weizenbaum, Joseph (January 1966), "ELIZA—A Computer Program For the Study of Natural Language Communication Between Man And Machine", Communications of the ACM 9 (1): 36–45, doi:10.1145/365153.365168.

[13] Terry Winograd, "Procedures as a Representation for Data in a Computer Program for Understanding Natural Language", MIT AI Technical Report 235, February 1971