

Voice Controlled Database Analysis

Varun Wahi
Computer Engineering, MITCOE
Pune, India

Prof. Avadhoot Joshi
Computer Engineering, MITCOE
Pune, India

Rohan Patel
Computer Engineering, MITCOE
Pune, India

Asad Mujawar
Computer Engineering, MITCOE
Pune, India

Nishant Tayde
Computer Engineering, MITCOE
Pune, India

Abstract— It is difficult for use having no technical knowledge to access a system. One needs to have the knowledge regarding formal languages to access information from current systems and this hinders non-technical people from obtaining the information they want. In this case artificial intelligence can increase simplicity and accessibility of a system for non-technical user. It is crucial for systems to be user-friendly in order to obtain the highest benefits. These systems try to make information accessible to everyone who knows a natural language. The main motivation of proposed systems is to break the barriers for non-technical users and make information easily accessible to them. Making a user friendly and more conversationally intelligent system will help user and even naïve users to perform queries without having actual knowledge of SQL or database schema. For instance, consider that a non-SQL user wants to retrieve information from database. The user is not acquainted with the database and SQL commands, making this task difficult. The proposed system takes such problems into consideration and provides solution to these problems. With natural language as input and conversion of natural language to SQL queries, even naïve users can access the data in database.

Keywords— *SQL, user-friendly, information retrieval, natural language processing.*

I. INTRODUCTION

Use of databases is widespread. Databases have applications in almost all information systems such as transport information system, financial information system, human resource management system etc.

Structured Query Language (SQL) queries get increasingly complicated as the size and complexity in the relation among the entities increase. These complex queries are very difficult to write for laymen or users who do not know SQL. The main problem here is that users who want to get information from the database do not know formal languages like SQL. A user is required to know all the details of the database such as relations, entities etc. Natural language interface to database presents an interface for non-expert users to interact with the system and database. To design models for automatically mapping natural language semantics into programming languages has always been a major and interesting challenge in Computer Science. For example, accessing a database requires the knowledge of SQL and machine readable instructions that common users do not know. Ideally, they should only ask questions in natural language without knowing either the underlying database schema or any complex machine language. Questions entered in natural language form are translated into a statement in a

formal query language. Once the statement is formed, the query is processed by the DBMS in order to produce the required data. Databases respond only to standard SQL queries which are based on relational algebra. It is nearly impossible for a layman to be well versed in SQL querying as they may be unaware of the structure of the database- namely tables, their corresponding fields and types, primary keys and so on.

Providing a solution to this problem, this system has been proposed that will use natural language speech through voice recognition, convert natural speech to SQL query and display the results from the database.

II. PROPOSED SYSTEM

The user will give input in the form of speech or text. If it is speech, the voice input will be given to the Speech to text Converter and Communicator which converts it in the text form. Google Speech Recognition API can be used for conversion. The user will be able to analyze the text and update it manually if required. This natural language question (NLQ) will then be converted into a stream of tokens with the help of tokenizer and a token id will be provided to each word of the NLQ. The Parts of Speech tagger (POS) will generate a list of POS tags for each corresponding token in NLQ. We remove the stopwords, i.e. Words which don't contribute any meaning towards Query Formation. Our system builds a lexicon set from Thesaurus which contains synonyms of words and SQL dictionary which contains Keywords used in SQL syntax. It also creates an Attribute Map from the metadata of the database. Attribute Map will contain a description of all Tables along with all columns (Tablenames, Columnnames) and mappings between tables according to foreign keys. Keyword Extractor uses Lexicon Set and Attribute Map to identify and extract keywords which are either present in Lexicon Set or Attribute map. Each keyword is assigned a tag based on its position in the input as well as the category of the keyword. This forms our Meaningful Representation of Input data. Query Generation is performed in the form of a syntax tree. SQL commands have a syntax tree associated with them, which contains Nodes like Select Clause Node, Where clause node, etc. Next step is identification of the nodes present in our NLQ. We analyze MR to identify QueryNodes. This identification is carried out using detection of SQL keywords present in the NLQ (SQL keywords are present in Lexicon set). After identification of nodes, we identify the relations between (Keyword,Nodes) and (Keyword, Other Keywords). Here we detect the columns, tables, operators, values and other SQL

clause specifics and assign them to their QueryNodes. Finally we generate Query by combining all the QueryNodes in order as per SQL syntax tree.

Tagging and Keyword Extraction Module: Natural language input is tagged using POS tagger. We used the POS tagger of Standard Stanford NLP Library, for parsing the input. The output of tagging step is a list of tokenized words and their POS tags specifying whether it is noun, proper noun, adjective, verb or other. From this list we will extract nouns, adjectives, proper nouns, numeric values etc. In this step, stop word elimination is also carried out.

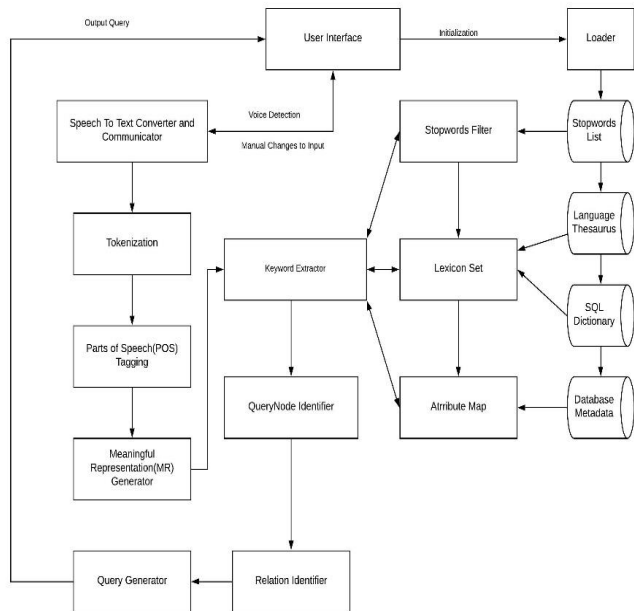


Fig 1: Proposed System Architecture

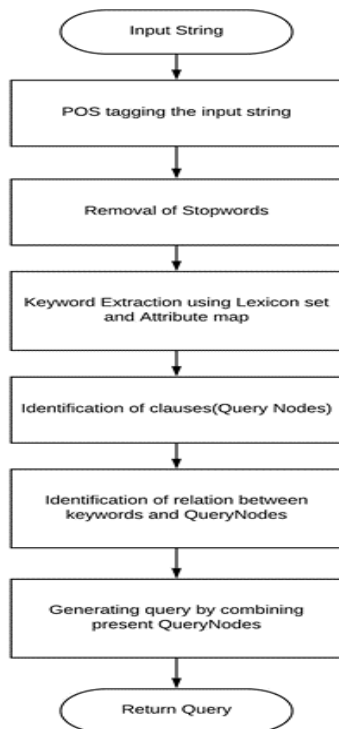


Fig 1.2: Query Generation Algorithm

Queries supported by proposed system:

- Select clause
- Where clause with all operators
- Limit clause
- Order by clause
- Group by clause
- Distinct clause
- Count, average, sum, minimum and maximum clause

III. MATHEMATICAL MODEL

Parsing and Keyword Extraction Module: Natural language input is parsed using parser. Parsers such as Stanford parser and OpenNLP can be used for parsing the input. The output of parsing step is a parse tree of given input which contains POS tags along with each word of input specifying whether it is noun, proper noun, adjective, verb or other. From parse tree we will extract nouns, adjectives, proper nouns, numeric values etc. In this step, stop word elimination is carried out.

MR Generation: The next step is to use Hyperspace Analog to Language matrix (HAL) i.e. word co-occurrence matrix algorithm to find out nouns related to proper nouns, adjectives and numeric values from given input.

Similarity between two vectors is calculated using Cosine Similarity. Consider two vectors for two words to calculate cosine similarity, say A and B then, cosine-similarity is represented as,

$$\text{Cosine Similarity (A, B)} = \frac{A \times B}{||A|| \times ||B||}$$

$$= \frac{\sum_{i=1}^n A_i \cdot B_i}{\sqrt{\sum_{i=1}^n A_i^2} \sqrt{\sum_{i=1}^n B_i^2}}$$

Find pairs having highest cosine-similarity value for any proper noun or adjective or numeric value. This is how we can get nouns associated with proper nouns, adjectives and numeric value from input. If we get 'proper noun-table name' pair as highest cosine-similarity value pair then search for second highest pair to get correct 'proper noun- noun' pair from input. The 'adjective-noun' pair is useful in generation of queries related to aggregate functions. The 'proper noun-noun' pair or 'numeric value-noun' pair is useful in generation of WHERE clause. E.g. Show address of employee whose salary is 20000. Use of module stated above gives us '20000salary' pair as 'numeric value-noun' pair and we can use it in WHERE clause as 'WHERE salary= 20000'.

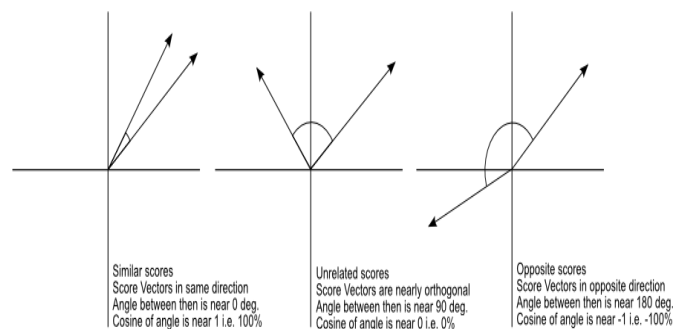
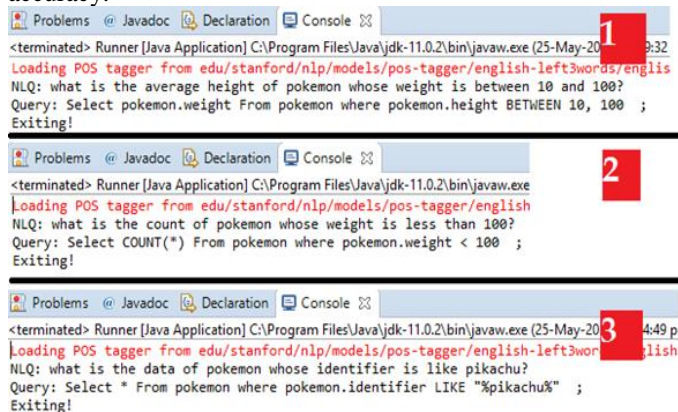


Fig 2: Cosine similarity vectors

IV. RESULT

The system was subjected to tests and the outputs were documented against the expected outputs. Inputs were restricted to the queries mentioned in the proposed system. The outputs obtained matched the expected output with 94.63% accuracy.



```
1
<terminated> Runner [Java Application] C:\Program Files\Java\jdk-11.0.2\bin\javaw.exe (25-May-2019) 9:32
Loading POS tagger from edu/stanford/nlp/models/pos-tagger/english-left3words/english
NLQ: what is the average height of pokemon whose weight is between 10 and 100?
Query: Select pokemon.weight From pokemon where pokemon.height BETWEEN 10, 100 ;
Exiting!

2
<terminated> Runner [Java Application] C:\Program Files\Java\jdk-11.0.2\bin\javaw.exe
Loading POS tagger from edu/stanford/nlp/models/pos-tagger/english
NLQ: what is the count of pokemon whose weight is less than 100?
Query: Select COUNT(*) From pokemon where pokemon.weight < 100 ;
Exiting!

3
<terminated> Runner [Java Application] C:\Program Files\Java\jdk-11.0.2\bin\javaw.exe (25-May-2019) 4:49 p
Loading POS tagger from edu/stanford/nlp/models/pos-tagger/english-left3words/english
NLQ: what is the data of pokemon whose identifier is like pikachu?
Query: Select * From pokemon where pokemon.identifier LIKE "%pikachu%" ;
Exiting!
```

V. CONCLUSION

Intelligent Query System using Natural Language Processing is a system used for making data retrieval from database easier and more interactive. Proposed system is bridging the gap between computer and casual user. Without any technical training handling databases is not possible for naïve user. This drawback is overcome by this system. This system converts the human speech input i.e. natural language input to the SQL query after converting the natural language to SQL query the generated query is given to database which gives the desired output. This report gives an overview of the successes and shortcomings of the system.

VI. FUTURE SCOPE

In proposed system, the process of natural language queries is independent of each other. Search is not often a single-step process. A user may ask follow-up questions based on the results obtained. It is thus necessary to provide a system to support a sequence of related queries. In the future, we would

like to explore how to support follow-up queries, thereby allowing users to incrementally focus their query on the information they are interested in, especially in conversation-like interactions. We would also aim at adding joins and inner clauses in the future iterations of this system.

VII. ACKNOWLEDGMENT

It gives us great pleasure in presenting the preliminary project report on 'Voice Controlled Database Analysis'. With due respect and gratitude we would like to take this opportunity to thank our internal guide Prof. Avadhoot Joshi for giving us all the help and guidance we needed. We are really grateful for his kind support. He has always encouraged us and given us the motivation to move ahead. He has put in a lot of time and effort in this project along with us and given us a lot of confidence. We are also grateful to Prof. Bharti Dixit, Head of Computer Engineering Department, MIT College of Engineering for its indispensable support. Also, we wish to thank all the other people who have helped us in the successful completion of this project. We would also like to extend our sincere thanks to Principal Dr. Anil S. Hiwale, for his dynamic and valuable guidance throughout the project and providing the necessary facilities that helped us to complete our dissertation work. We would like to thank our friends who have helped us directly or indirectly to complete this work.

VIII. REFERENCES

- [1] Fei Li, H.V. Jagadish, "Constructing an interactive natural language interface for relational database" Journal proceedings of VLDB endowment, vol. 8, Issue 01, Sept.
- [2] Probin Anand, Zuber Farooqui, "Rule based Domain Specific Semantic Analysis for
- [3] K. Javubar Sathick, A. Jaya, "Natural Language to SQL Generation for Semantic
- [4] Tanzim Mahmud, K. M. Azharul Hasan, Mahtab Ahmed, "A Rule Based Approach
- [5] Enikuomehin A.O., Okwufulueze D.O, "An Algorithm for Solving Natural Language Query Execution Problems on Relational Databases" (IJACSA) International Journal of Advanced Computer Science and Applications, Vol. 3, No. 10, 2012.
- [6] Sachin Kumar, Ashish Kumar, Dr. Pinaki Mitra, Girish Sundaram, "System and Methods for Converting Speech to SQL" Appeared in proceedings of International Conference on "Emerging Research in Computing, Information, Communication and Applications" ERCICA 2013.