e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:06/Issue:07/July-2024 Impact Factor- 7.868 www.irjmets.com

VOICE-ACTIVATED SQL QUERY GENERATION

Varun VM*1

*1PES Institute Of Technology And Management, Shimoga, Karnataka, India.

ABSTRACT

The growth of various artificial intelligence (AI) and natural language processors (NLP) made it possible to produce systems that convert spoken words into actual SQL database queries. Techniques of Automatic Speech Recognition (ASR) involving the use of Python's SpeechRecognition package and Google Speech Recognizer API are utilized in deep learning models to analyze and synthesize intricate SQL structures. These models, trained by countless examples of compound queries, achieve users' goals with databases. This concept is easily understandable and enables any layperson or technically challenged individual to interact with and organize their data. As an AI-controlled technique, it is best used for business analysis and personal data administration, allowing users to work with databases using only their voice.

Keywords: Natural Language Processors (NLP), Automatic Speech Recognition (ASR).

I. INTRODUCTION

The integration of AI-driven SQL queries Using voice commands is a big improvement in human-computer interaction, combining the capabilities of automated speech recognition (ASR) and natural language processing (NLP) with database management systems. This innovation attempts to make data searching more accessible by allowing users who are new with SQL syntax to interface with databases via simple voice commands. ASR systems transform spoken words into text, utilising advances in deep learning to improve accuracy. These systems can handle a types of accents, dialects, and speech patterns, making them appropriate for a wide range of user groups. NLP technologies interpret and comprehend human language, allowing for the translation of spoken or written language into structured SQL commands. This includes interpreting the input, determining the user's purpose, and creating the proper SQL query. Modern AI systems use sophisticated models like sequence-to-sequence (Seq2Seq) models to convert voice commands into SQL queries. These models are trained on large datasets to recognise patterns and nuances in spoken language, allowing them to construct complicated SQL queries that include joins, nested queries, and more. This technology has various advantages. It promotes usability by decreasing the barrier to database interaction, making it available to non-technical people. In corporate situations, swift data retrieval and analysis are critical, and users can perform data queries more efficiently by speaking rather than typing, saving time and reducing the possibility of errors. Furthermore, The accessibility for those with impairments is enhanced by this technology, giving them another way to engage with databases. Despite these advantages, there may still have few problems. It is hard to ensure that the system accurately understands and processes the user's purpose, especially when dealing with ambiguous or complex questions. The system must be strong enough to handle a wide range of questions, from simple data retrieval to complicated analytical inquiries. Furthermore, combineing this technology with available database management systems and assuring scalability are critical for its wider use. The future of AI-driven SQL queries from voice commands lies in overcoming these challenges to provide robust and reliable solutions for diverse linguistic and cultural contexts. Ongoing research and development in ASR and NLP technologies will play a important role in achieving this goal, ensuring more reliable and comprehensive solutions for users worldwide.

II. LITERATURE REVIEW

"A Model of a Generic Natural Language Interface for Querying Database,"[1] Bais et al. (2016) which proposes a system that allows user to query databases using natural language. The model integrates linguistic and database knowledge components to interpret and process user queries effectively. It leverages artificial intelligence techniques to handle natural language input, improving the interface's usability for database interactions.

"Automatic SQL Query Formation from Natural Language Query,"[2] Ghosh et al. (2014) present an NLP systems that change natural language queries into SQL commands. The technology forms SQL queries correctly

e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:06/Issue:07/July-2024 Impact Factor- 7.868 www.irjmets.com

by using phases like morphological, lexical, syntactic, and semantic analysis.This approach aims to simplify database interactions by enabling users to retrieve data using plain language rather than complex SQL syntax.

"Automatic SQL Query Formation from Natural Language Query," [3] Nagare et al. (2017) describe a system which converts text or speech in normal language into SQL queries. The system validates the queries, breaks them into tokens, and compares them with stored clauses to construct and execute SQL commands. This method simplifies database querying, making it accessible to users without technical SQL knowledge.

"Extracting SQL Query Using Natural Language Processing,"[4] Nandhini S et al. (2019) focus on using NLP techniques to create SQL queries in natural language input. Their implementation involves stages such as query creation, lexical analysis, syntactic analysis, and semantic analysis.This approach aims to make database querying more intuitive and accessible by allowing users to communicate with databases using everyday language.

"Natural Language Processing with some Abbreviation to SQL"[5] Chandrakala Kombade et al. (2019) present a system that utilizes NLP to convert natural language input, including abbreviations, into SQL queries. Their work involves parsing the input, understanding the context, and generating appropriate SQL commands. This approach enhances the usability of database systems by enabling more natural and flexible user interactions.

III. METHODOLOGY

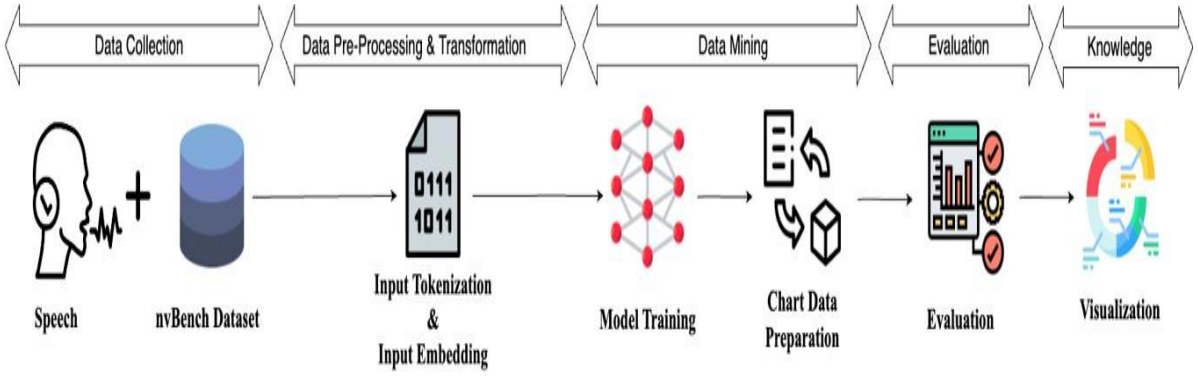

Figure 1. Process Flow of Methodology

3.1 Data Collection

The process of gathering the trained data needed for our deep learning models is called as data collecting. Comprehensive voice datasets tailored to natural language inquiries pertaining to visualisations are not yet accessible. As a result, we rely on speech recognition technology to translate spoken words into text. We selected the nvBench1 dataset for this work since it is designed for cross-domain activities that involve questions and visualisations in natural language. The collection has over 25,000 pairs of combinations (natural language, visualisation) in 105 different categories, encompassing several visualisation styles and SQL queries that yield table-valued outcomes.

3.2 Data Pre-processing & Transformation

To improve the'source' column in data pre-processing, initialization tokens such as "<sos>" (sentence start) and "<eos>" (sentence end) are included for efficient model training. Data is divided into tokens via input tokenization, which includes "<NL>" for natural language questions, "<C>" for sketching SQL queries, "<D>" for database information, "<COL>" for columns, and "<VAL>" for values. Then, motivated by current research, these tokens are transformed into vectors using token embedding (individual token representation), type embedding (token type identification), and position embedding (token sequence position marking).

3.3 Data Mining

In this step, we make use a Sequence-to-Sequence (Seq2Seq) model, which is an advanced sort of Recurrent Neural Network (RNN) that includes both encoder and decoder components. These components use several self-attention blocks to efficiently process and produce sequences. The encoder accepts processed input embeddings and starts with a "<sos>" identifier to indicate the beginning of the embeddings. The decoder then outputs the output sequence, which predicts the SQL query based upon the encoded input. The iterative method continues until a "<eos>" token, signifying the end of the phrase, is anticipated. Attention Forcing and

e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:06/Issue:07/July-2024 Impact Factor- 7.868 www.irjmets.com

SQL-aware translation techniques help the model create SQL components more precisely, such as select columns, aggregate functions, table names, conditions, groups, sorting, limits, and chart kinds.

3.3.1 Chart Data Preparation for Visualization

Once a valid SQL query is generated, it is executed on an SQLite3 database to retrieve structured data in rows and columns. The resulting data is stored in a pandas dataframe. Depend upon the predicted chart type from the natural language question, specific visualization functions (e.g., draw_bargraph(), draw_linegraph()) are invoked using matplotlib, a Python package for creating visualizations.

3.4 Evaluation

Evaluation in the model's performance is critical for determining its correctness and efficacy. Quantitative assessment measures how well a produced SQL queries match the ground truth SQL queries. This accuracy indicator, defined as an ratio of correct SQL queries (P) to total number of testing instances (Q), gives a quantitative assessment in the model's ability to provide valid SQL outputs. Furthermore, execution accuracy assesses the syntactical correctness of created SQL queries. Ensuring accurate syntax is critical since mistakes in syntax might cause issues when running queries on the SQLite3 database are any other database management system. Researchers and developers may completely examine the model's capabilities and discover areas for improvement in producing SQL commands using inputs in natural language.

3.5 Knowledge

In this final stage, knowledge is extracted by fulfilling the user's request for visualization based upon their natural language query. Data extracted from the SQL query is formatted into a dataset suitable for chart display. Using pandas dataframes, the requested chart type guides the selection of appropriate visualization methods, such as draw_bargraph() or draw_linegraph(), leveraging matplotlib for visualization creation.This comprehensive process aims to bridge the space between natural language queries about visualizations and actionable SQL queries and visual representations, supporting diverse applications in data analysis and decision-making.

IV. SYSTEM DESIGN

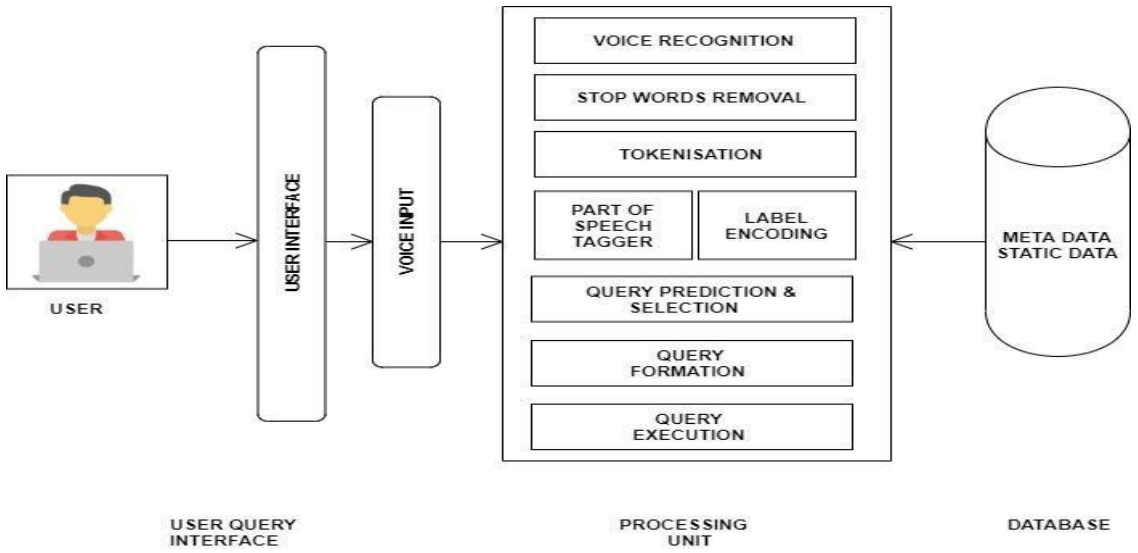

Figure 2: System Architecture

1. User: The person interacting with the system and entering data via a graphical user interface (usually by voice commands).

2. User Interface: The user communicates with the system via this medium. The user's vocal input is captured.

3. Voice Input: After processing, the speech input is turned into text for additional study

4. Voice Recognition: This part converts the oral input to written language. Speech recognition algorithms are utilised to comprehend and record spoken phrases by the user.

5. Stop Words Removal: This method simplifies the text for processing by eliminating frequent terms (such as "the," "is," and "at") that don't significantly add to the content in a question.

e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:06/Issue:07/July-2024 Impact Factor- 7.868 www.irjmets.com

6. Tokenization: In this phase, the text is divided into discrete words, or tokens. Tokenization is essential to comprehending the query's structure.

7. Part of Speech Tagger: This phase examines the tokens to determine which grammatical components—nouns, verbs, and adjectives—they include. Comprehending the functions of words facilitates the creation of precise inquiries.

8. Label Encoding: Through this procedure, the text input is transformed into a numerical representation that the query prediction model can simply process. There is a unique number label allocated to every word or token.

9. Meta Data Static Data: The system leverages the structured data and info in this database to evaluate and provide precise queries. By giving the processing unit the required reference data, it supports it.

10. Query Prediction & Selection: The system guesses which query the user is most likely to conduct based on the input that has been evaluated. It chooses from among possible queries the best match.

11. Query Formation: Based on the chosen forecast, the real SQL query is created at this step. The tokens and labels are arranged into a syntactically sound SQL query by it.

12. Query Execution: To get the needed data, the last SQL query is run against the database. The user is then presented with the results via the interface.

V. RESULTS AND DISCUSSION

The natural language translation is enabled to turn voice instructions into SQL to assist individuals who cannot interact with a database using SQL. Some studies indicate that first-level tests give accuracy of more than 73%, which greatly enhances the efficiency of queries as well as user friendly nature.

Figure.3: Home Page

e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:06/Issue:07/July-2024 Impact Factor- 7.868 www.irjmets.com

Figure.4: Give Input Through Voice Commands

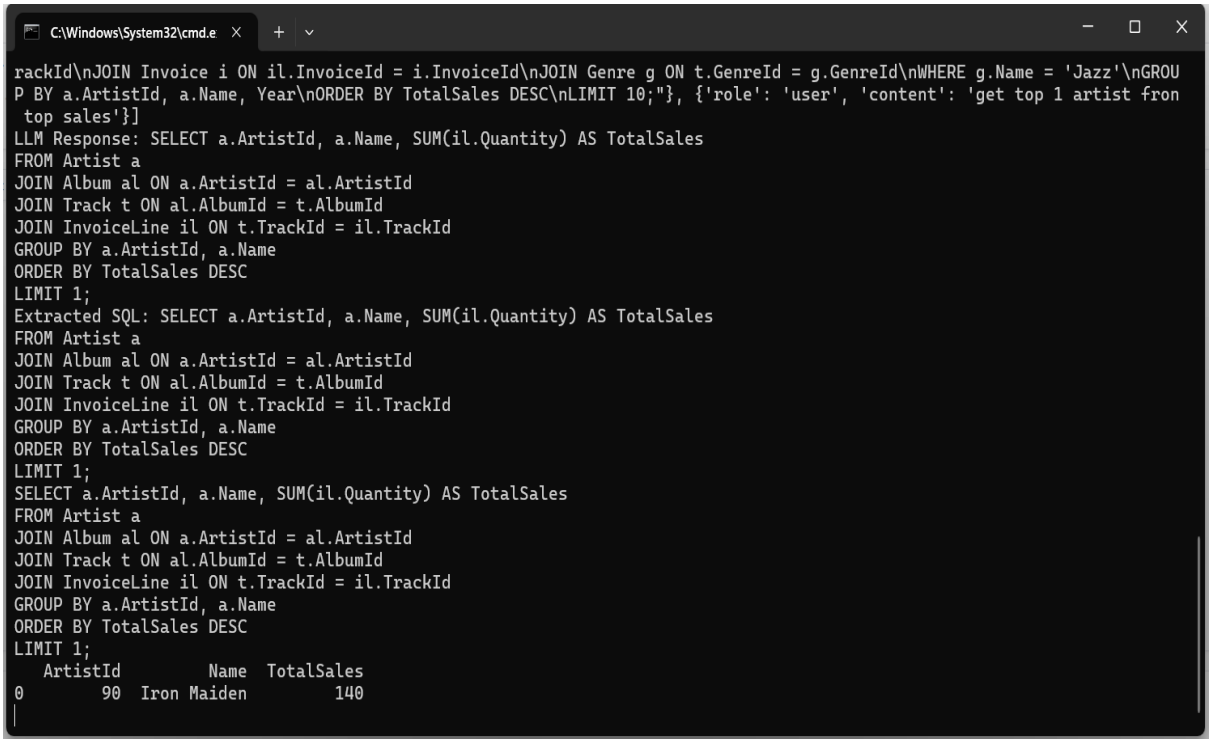

Fig.5: Output In Cmd

VI. CONCLUSION

Artificial intelligence and ML have dramatically improved many people's lives. Our findings also included an approach for improving the coding experience. The project successfully achieves its purpose of supporting people suffering from Stress ,Injury and individuals with no coding knowledge. The model provides accurate and efficient queries for user who enter natural language, simplifying data retrieval. This method can benefit a wide range of companies, including those in education and healthcare. To maintain peak performance, update the database on a regular basis with system-specific phrases.

e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:06/Issue:07/July-2024 Impact Factor- 7.868 www.irjmets.com

VII. REFERENCE

[1] Bais, Hanane, and Lahcen Koutti. "A Model of a Generic Natural Language Interface for Querying Database." In international journal of intelligent system and applications. 8. 35-44.10.5815/ijisa.2016.02.05.

[2] Ghosh, Prasun Saltlake, Kolkata Kolkata, Saparja Dey, Kolkata Sengupta, Subhabrata Assistant,Kolkata Saltlake,. (2014). "Automatic SQL Query Formation from Natural Language Query", Inter_national Conference on Microelectronics, Circuits and System (MICRO-2014).

[3] Nagare, Indhe, Sabale, Thorat and Chaturvedi. "Automatic SQL Query Formation from Natural Language Query" International Research Journal of Engineering and Technology (IRJET) Mar -2017

[4] Nandhini S, B.Viruthika, Almas Saba, Suman Sangeeta Das "Extracting Sql Query Using Natural Language Processing" International Journal of Engineering and Advanced Technology (IJEAT) April 2019

[5] Chandrakala Kombade, Monika More, Shweta Patil, Anjalidevi Pujari "Natural Language Processing with some Abbreviation to SQL" International Journal for Research in Applied Science Engineering Technology (IJRASET) Dec 2019.