Grenze International Journal of Engineering and Technology, June Issue

Machine Learning-Powered Voice-based SQL Querying System

Savita V Kandharkar[1] and Rushali Deshmukh[2]
[1-2]Pune University, Computer Engineering, JSPM's, Rajarshi Shahu College of Engineering
Email: kandharkarsavita@gmail.com, radeshmukh_comp@jspmrscoe.edu.in

***Abstract*— We designed this system to allow safe and efficient access to data via a voice-controlled interface, incorporating speech-based user verification and smooth alignment with a pre-existing NLP framework. We explore techniques to improve the retrieval process from query outputs. Traditional fixed query structures are inadequate for handling various ad-hoc user searches within databases. In this work, we propose a machine learning-driven method for generating SQL queries based on voice commands, along with an advanced interface for dynamically creating database query forms.**

***Index Terms*— Speech recognition, Natural Language Processing (NLP), Query optimization, Machine learning, Database interaction, Voice authentication, Dynamic query generation, Microcode synthesis, Compiler technologies.**

I. Overview

While natural language is the most intuitive method for human understanding and interaction, it presents significant challenges for machine interpretation. The goal of Natural Language Processing (NLP) seeks to bridge the gap between human language and machine understanding, removing the necessity for users to remember complex instructions. NLP allows machines to interpret human languages in a way that feels natural. This project is centered on transforming user questions in English into SQL statements by applying semantic grammar rules. The platform processes natural language input from users, checks the query's correctness, and decomposes the input into separate tokens, where each token corresponds to a word from the query. These tokens are matched with pre-existing clauses in a dictionary, which is continuously updated. The system then processes the tokens to identify the attributes referenced in the query. By comparing the syntax and semantics, we determine which tables in the database contain the relevant attributes. The process concludes by creating the Query instruction, performing its execution on the Storage backend, and returning the results to the user.

II. Literature Survey

The device supports bidirectional natural language interaction through both text and speech. Communicating through written language tends to be simpler than using speech. In Text-based communication, analysis is conducted on syntax, semantics, lexicon, and morphology. Verbal communication encompasses all writing processes, along with supplementary knowledge of phonology and other information necessary to address the uncertainties inherent in speech [1].

Grenze ID: 01.GIJET.11.2.572_26

This research emphasizes advancing the ability to progressively understand and interpret speech within virtual conversational agents. The proposed project contributes to a continuing initiative designed to facilitate effective spoken interaction with virtual individuals in multiparty arbitration contexts. Trainers may refine their intervention abilities in these settings by participating in direct verbal negotiations with one or more virtual individuals. An essential aspect of enabling virtual people to behave realistically in negotiating scenarios is their comprehension of turn-taking, logic, trust, and emotions [2].

Virtual human interaction platforms commonly rely on refined human audio samples or speech synthesis technologies tailored for particular subject areas. Both methods result in an aesthetically acceptable performance, but at a higher expense. To find the right balance with respect to functionality and cost, An analysis is conducted on human and generated speech based on perceived realism, conversational qualities, as well appeal. We adjust the type, length, and content of spoken expressions, considering the evaluators' age group, first language, and expertise in speech synthesis. The results suggest that a skilled human voice tends to perform better than novice individual audio and synthetic speech. Additionally, A superior flexible voice or a well-designed area-specific audio could surpass occasional human recordings. As expected, the restricted-domain voice often earns worse evaluations compared to the high-quality general-purpose voice. A non-statistically significant tendency of decreased ratings for lengthy or unpleasant statements has also been found [3].

This study aims to explore the commercial use cases of chatbots and propose various criteria to evaluate their effectiveness, user experience, and overall performance as intelligent conversational agents. Using these criteria, we evaluate current commercial chatbots in the Polish language, primarily functioning within the B2C sector. To continue, engage the broadest potential spectrum of consumers. Ultimately, these systems represent advanced commercial applications developed by their creators. The system looks at many aspects of how each chatbot performs: how they look, how they work on the website, how they talk, what information they have, how they share that information, their conversation skills and understanding of context, their personality traits, options for customization, how they handle emergencies, and how users can rate both the chatbot and the website [4].

This paper introduces a probabilistic framework for conversation systems structured around the Markov Decision Process (MDP) and proposes a structured quantitative model for human-computer conversational interactions. At the start, the researcher lays out an in-depth formulation of the systems, emphasizing their state space, action set, and strategic framework. In the context of this paradigm for Markov decision processes, it can be shown that the challenge of designing conversation strategies can be articulated as an optimization issue, which may be addressed using many methodologies, including reinforcement learning techniques. The author is developing first findings on a conversation method for a system that provides air travel information [5].

The paper suggests that in conventional information retrieval tasks, like retrieving documents, errors in speech recognition are usually either minimal or solvable through simple approaches. The author has mitigated it to some extent by acknowledging such recognition mistakes. The author is developing voice recognition technology with user input. In the absence of sufficient vocabulary, there is no possibility for redundancy or contextual cues to mitigate inaccuracies [6].

The deployed system is fundamentally a college-focused virtual assistant. The established system addresses student inquiries about college-related matters. The developed system employs an authentication technique identity purpose. The verification system incorporates password security. SQLite apply for user credential protection. We developed the software using Android IDE [7].

The developed application functions as a virtual assistant specifically tailored for college environments. It responds to student queries related to academic and campus-related topics. To verify student identities, the system employs a secure login mechanism that includes password-based validation. User credentials are securely managed using an SQLite database. The application was built using the Android development environment. [8].

The physical device captures the spoken input through the built-in microphone and interprets it to deliver a response using the embedded audio output unit. For example, if the user asks, "What events are scheduled today?" or "When is my next class?", the system utilizes its integrated features to retrieve the necessary information and communicates the results to the user via the connected speaker. [9].

This article proposes a method for creating a personal assistant that minimizes reliance on input devices such as mice and keyboards for personal computers. Issuing orders via verbal communication enhances user-friendliness. This study delineates the representation model, the personal assistants currently available on the market, and how this system is implemented. The study also highlights the added functionality of remote control and the inclusion of novel commands that distinguish it from other solutions [10].

This study introduces an innovative methodology for intelligent search. A significant number of individuals worldwide use virtual assistants. The study discusses the uses of virtual assistants that provide opportunities for

mankind across numerous sectors. This research also elucidates the need to use virtual assistant technologies [11].
This article introduces a method for automatically generating a whitelist using queries created during web application testing. Author's suggested approach utilizes the questions produced during application testing. It is not dependent upon particular applications, resulting in enhanced responsiveness to threats and adaptability for various applications [12].

III. PLANNED ARCHITECTURE

During the first phase, voice data is gathered and archived for use in the following processing stage. In the subsequent phase, the audio data is consistently analyzed and transcribed to text. Subsequent step involves analyzing and processing the transformed text using a Python script to determine the appropriate answer to the command. Ultimately, upon identifying the answer, output is produced depending on the database's creation.
The architecture shown below represents the system. The diagram above illustrates the fundamental operation of the system. The user submits either textual or vocal input. The speech API transcribes spoken instructions into written text. The input text is maintained in the database for additional processing steps. There is a correlation between it and the database's instructions. After locating the command, the user interface outputs the corresponding task as text.

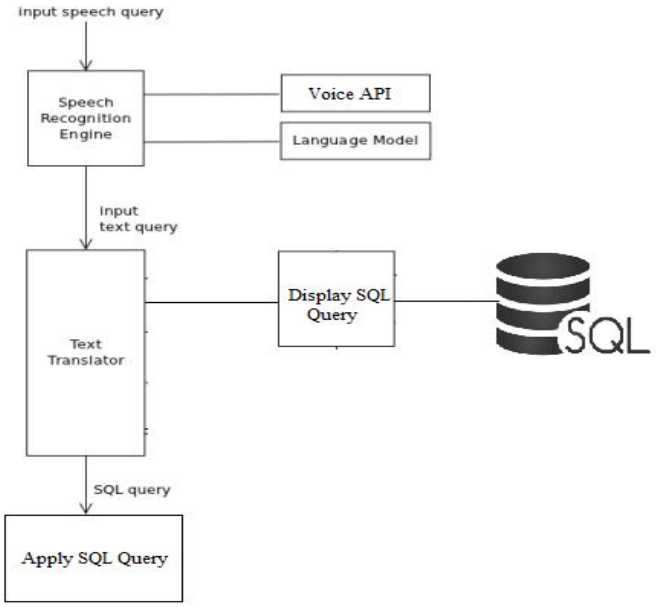

Fig 1. System architecture

IV. SYSTEM FLOW

- Input Speech Query: The user initiates a query by speaking a question or command. This audio input is captured by the application.
- Speech Recognition Engine: The captured speech is processed by the Speech Voice API (e.g., Google Speech Recognition API) to convert speech into text. A language model is used to understand context and improve transcription accuracy.
- Text Translator: The recognized text (natural language query) is passed to the Text Translator. This component converts the plain English query into an equivalent SQL query using tools like ln2sql (a language-to-SQL translation model), executed via a Python subprocess to generate SQL from the input text.
- Display SQL Query: The generated SQL query is displayed to the user. This step adds transparency and allows users to verify that the SQL query matches their intent.
- Apply Database Structured Query: The validated database query is sent to the backend SQLite database (school.db).
- Fetch The Results: The results are fetched and displayed in a neat, tabular format on the web page.

V. System Components

A. Core Functionality

- The system assists non-technical users in learning SQL queries through voice and text interaction.
- Users can submit their own queries to a demo database using either voice or text input.
- Users have the option to search for their queries on Wikipedia.

B. Platform Model Working

- Flask – Used as the web framework to build the application's user interface and route HTTP requests.
- Speech Recognition + Google API – Implements speech-to-text conversion, allowing users to input queries via voice.
- Subprocess + ln2sql – Converts natural language queries into SQL statements using the ln2sql tool via a subprocess.
- SQLite (school.db) – Serves as the backend database to store and manage school-related data.
- Pandas – Executes the generated SQL queries and displays the result.

VI. Computational Formulation

Input-Response:
U= {I, R, f, S, F}
Where,
I (Input Set):
I= {I1, I2}

- I1 : Set of textual SQL queries {l1, l2 ,...,ln}where each li is a SQL query input.
- I2: Voice-based SQL query input obtained from speech recognition.

R (Output/response Set):
R= {R1,R2,R3, R4, R5}
R1= Audio data preprocessing
R2= query generation
R3 =query execution
R4 = query validation and error handling
R5 = Voice-based response creation
f (Functions):
f= {f1,f2}
f1= Preprocessing function for voice and text inputs
f2 = Query analysis and processing function
S (Success Condition):
SQL query is accurately generated, executed, and produces the expected response.
F (Failure Condition):
Query generation or execution fails due to algorithm errors or voice recognition issues.

VII. Evaluation

Displays the recognized speech after converting it to text using the Google Speech API. The system displays the generated SQL query after converting the spoken (natural language) input into SQL using the ln2sql function.

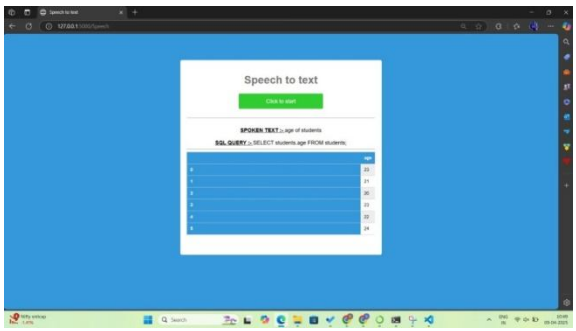

Fig 2. Speech To Text

This confirms that the Speech Recognition + Google API successfully captured and converted the voice input into text. SELECT students.age FROM students; The result is the translated SQL query generated by the ln2sql tool via a sub process, based on the spoken input. It indicates that the system correctly interpreted the user's request. The system provides a clear, tabular output that displays the ages of the students retrieved from the school. db SQLite database. The table is likely rendered using Pandas, which formats SQL results neatly for display within the Flask web app. Following Fig3, Fig4 and Fig5

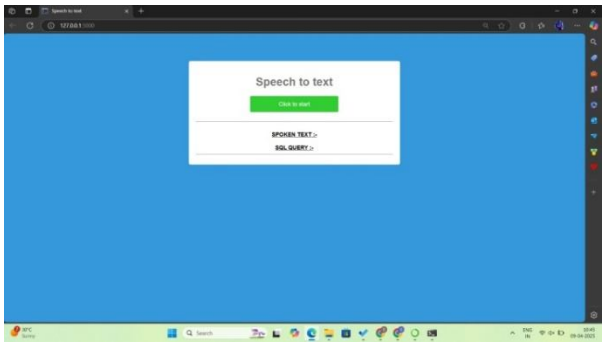

Fig 3. Age of Student Result

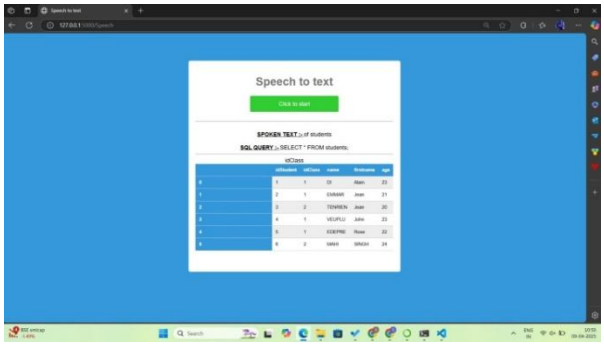

Fig 4. Class of Student Result

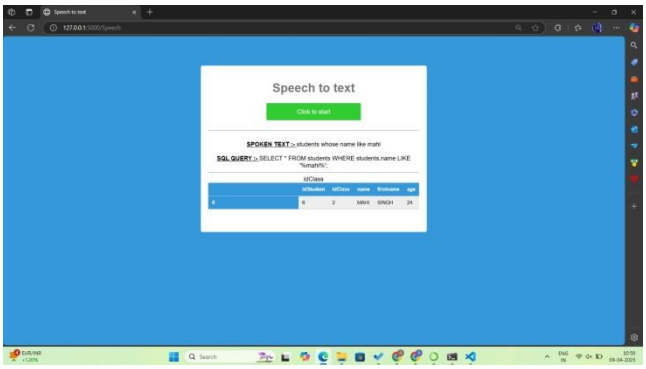

Fig 5. Search Student Name Result

VIII. Conclusion

Using everyday communication facilitates convenience for individuals. Proposed technology facilitates users in effortlessly extracting data from the database using plain English This technology allows users to effortlessly access data from the database through natural language, removing the requirement to learn complex query languages such as SQL. To further improve the system's ability to handle a larger volume of queries, we plan to incorporate additional synonyms for table and column names. The system further archives the successfully run queries derived from voice generation. This system provides suggestions to improve the user's experience. In the future, we may include a robust recommendation structure in this system to minimise user effort.

References

[1] F.Siasar djahantighi, M.Norouzifard, S.H.Davarpanah, M.H.Shenassa, "USING NATU- RAL LANGUAGE PROCESSING IN ORDER TO CREATE SQL QUERIES", in Proceedings of the International Conference on Computer and Communication Engineering, Kuala Lumpur, May 2008, pp. 600-604.

[2] NATURAL LANGUAGE PROCESSING USING PYTHON International Journal of Scientific Engineering Research Volume 8, Issue 5, May-2017 19 ISSN 2229-5518[IEEE]

[3] Natural Language Processing Techniques Applied in Information Retrieval-Analysis and Implementation in Python, TulikaNarang, International Journal of Innovations Advance- ment in Computer Science IJIACS ISSN 2347 – 8616 Volume 5, Issue 4 April 2016

[4] Anuradha Mohite, Varunakshi Bhojane, "Natural Language Interface to Database Using Modified Co-occurrence Matrix Technique 2015 International Conference on Pervasive Computing (ICPC)

[5] Levin E., Pieraccini R., Eckert W. "Learning dialogue strategies within the Markov decision process framework", Automatic Speech Recognition and Understanding IEEE Proceedings, pp. 72-79, 1997.[IEEE]

[6] Liddy, E. D. In Encyclopedia of Library and Information Science, 2nd Ed. Marcel Decker, Inc.
[7] J. Allan. Perspectives on information retrieval and speech. In Information Retrieval Techniques for Speech Applications: LNCS 2273, pages 1–10, 2002.
[8] Vishmita Yashwant Shetty, Nikhil Uday Polekar, Sandipan Utpal Das, Prof. Suvarna Pansambal, "ARTIFICIALLY INTELLIGENT COLLEGE ORIENTED VIRTUAL ASSISTANT Department of Computer Engineering, Atharva College of Engineering, Malad (W) Mumbai, India.
[9] Abhay Dekate, Chaitanya Kulkarni, Rohan Killedar, "Study of Voice Controlled Personal Assistant Device" International Journal of Computer Trends and Technology (IJCTT) – Volume 42 Number 1 – December 2016 ISSN: 2231-2803.