VoiceQuery System

Vijay Surya Vempati
University of Utah
Salt Lake City, Utah, USA
vijay.vempati@utah.edu

Sam Blesswin Stephen Rajan
University of Utah
Salt Lake City, Utah, USA
samblesswin.stephenrajan@utah.edu

1 GRADUATE-LEVEL PROJECT PROPOSAL

This Implementation project aims to develop a working demo of a voice query system that enables users to convert natural language questions into SQL queries using speech recognition and language understanding capabilities. Additionally, the project will include a comprehensive literature survey on natural language query systems and database management, focusing on the high-level theme of human-centered data management.

The project will involve developing a user-friendly interface using the React framework, integrating speech-to-text conversion functionality, utilizing large language model (LLM) APIs for natural language understanding, and interfacing with a PostgreSQL database for executing SQL queries.

2 LITERATURE SURVEY

Automatic Speech Recognition (ASR) is crucial in modern technology and communication, enhancing accessibility for people with disabilities and boosting productivity by enabling speech-based interaction. It facilitates hands-free operation, useful in driving or operating machinery, and supports multimodal interaction for natural user interfaces. ASR powers personal assistants like Siri and Alexa, streamlining tasks through voice commands, and enables language translation, improving international communication. It simplifies transcription for meetings and medical documentation and enhances customer service through interactive voice response systems.

The "Listen, Attend, and Spell (LAS)" [2] model is a seminal effort in the field of ASR, introducing a complete neural network design for sequence-to-sequence speech recognition. The major novelty of LAS is its attention mechanism, which enables the model to dynamically focus on significant sections of the input voice signal while creating the associated letter sequence. This attention mechanism allows LAS to align input voice frames and output characters, successfully modeling the temporal connections between them.It comprises an encoder network, an attention mechanism, and a decoder network. The encoder network converts the input speech signal to a series of high-level feature vectors. The attention mechanism computes a context vector at each decoding step, capturing the relevant information from the input.

Deep speech 2 [1] is another ground-breaking study in end-to-end ASR, proposing a deep learning model that achieves exceptional accuracy on both English and Mandarin voice recognition tasks. The model is built on deep bidirectional recurrent neural networks (RNNs) that were trained using the Connectionist Temporal Classification (CTC) method. Deep Speech 2's architecture comprises numerous layers of bidirectional RNNs, followed by fully connected layers for acoustic modeling. Unlike prior systems that rely on handmade features, Deep spoken 2 works directly with raw audio waveforms, allowing it to capture complicated patterns and relationships in the spoken stream.Scalability and efficiency are two of Deep Speech 2's most significant accomplishments. The model may be trained on huge datasets utilizing parallel processing, allowing for quick experimentation and deployment.

The Transformer Transducer model [6] is a new advancement in ASR that combines the benefits of transformer encoders with the versatility of the transducer framework. Unlike older ASR models, which frequently struggle with processing speed and real-time jobs, the Transformer Transducer is designed to operate in real-time, making it ideal for applications requiring speedy speech recognition.The model's architecture consists of transformer encoders for feature extraction and an RNN-based transducer network for sequence modeling and prediction. Transformer encoders excel at capturing long-distance connections in speech, and the RNN-T loss function aids in training and prediction efficiency.

Text-to-SQL is essential for simplifying database querying by allowing users to interact with databases using natural language. It democratizes access to data, enabling non-technical users to extract insights without needing SQL expertise. This automation boosts productivity and enhances data-driven decision-making across organizations.Seq2SQL [7] introduces a an approach to text-to-SQL generation using reinforcement learning (RL).Seq2SQL learns to generate SQL queries from natural language by treating the problem as a sequence-to-sequence task. By employing RL, Seq2SQL is able to optimize query generation over time, improving accuracy and robustness. This work represents a shift towards end-to-end approaches for text-to-SQL, enabling the model to learn directly from data without the need for handcrafted rules.EditSQL [3] addresses the challenge of generating SQL queries from natural language by proposing an approach based on sequence-to-sequence models with copy mechanism.It focuses on editing pre-existing SQL templates, which are then adapted to match the user's intent expressed in natural language. This approach improves query accuracy and reduces the complexity of the generation process. EditSQL leverages the strengths of both rule-based and data-driven methods, allowing it to handle complex queries while maintaining flexibility across different domains. The model has been evaluated on various benchmarks and has demonstrated competitive performance compared to state-of-the-art approaches

The importance of Speech-to-SQL lies in its ability to enable natural and efficient interaction with databases through spoken language, thereby reducing the barriers for users who lack expertise in SQL. By allowing users to query databases using voice commands or natural language questions, Speech-to-SQL systems enhance accessibility and usability, leading to improved productivity and user satisfaction. Speech-to-SQL [4] works towards designing more effective speech-based interfaces to query the structured data in relational databases. We first identify a new task named Speech-to-SQL, which aims to understand the information conveyed by human speech and directly

translate it into structured query language (SQL) statements.It proposes a novel end-to-end neural architecture named SpeechSQLNet to directly translate human speech into SQL queries without an external ASR step. SpeechSQLNet has the advantage of making full use of the rich linguistic information presented in speech. To the best of our knowledge, this is the first attempt to directly synthesize SQL based on arbitrary natural language questions, rather than a natural language-based version of SQL or its variants with a limited SQL grammar. These works represent important milestones in the development of natural language interfaces for database querying, offering insights and methodologies that contribute to the ongoing evolution of human-centered data management systems. Our implementation project takes inspiration from these works, aiming to build a practical voice query system hat incorporates similar natural language understanding and query generation techniques.

The latest demo is a VoiceQuerySystem [5] is a voice-driven database querying system designed to allow users to interact with databases using natural language questions (NLQs). Unlike existing systems like SpeakQL or EchoQuery, which require users to input exact SQL queries or follow predefined templates, VoiceQuerySystem enables data manipulation through common NLQs, eliminating the need for users to have a technical background in SQL.The system's underlying technique revolves around a new task called Speech-to-SQL, aimed at understanding the semantics in speech and translating it into SQL queries. We explore two approaches: a cascaded method and an end-to-end (E2E) method for speech-to-SQL translation. The cascaded method first converts the user's voice-based NLQs into text using a self-developed automatic speech recognition (ASR) module. Then, a text-to-SQL model (i.e., IRNet) generates SQL queries based on the converted text. In contrast, the E2E method introduces a novel neural architecture named SpeechSQLNet. This architecture directly converts speech signals into SQL queries without an intermediary text step.

3 TECHNICAL APPROACH

taking ideas from above VoiceQuerySystem [5]. We have implemented a demo of a voice query system. This is a cascaded approach, here are various layers.

(1) User Interface: Developed a responsive and intuitive user interface using React, incorporating features such as voice input, query results display, and feedback mechanisms.
(2) Speech-to-Text Conversion: Integrated a speech recognition library or API (e.g., google Speech to text API) to convert spoken words into text format, allowing users to input queries via voice commands.
(3) Natural Language Understanding (NLU): Utilized large language model (LLM) APIs such as OpenAI's GPT3.5 to understand the intent and entities of the user's query.
(4) SQL Query Generation: Develop logic to translate the parsed natural language queries into SQL queries. This involves mapping user intents to SQL operations and identifying relevant entities.
(5) PostgreSQL Integration: Interface with a PostgreSQL database to execute the generated SQL queries and retrieve results. Implement error handling to manage potential database errors or invalid queries. Timeline for the above is in Table **??** .

4 CONCLUSION

The project aims to deliver a functional demo of a voice query system that showcases the integration of speech recognition, natural language understanding, SQL query generation, and PostgreSQL database interaction capabilities. Users should be able to interact with the system using voice commands to input queries, which are then translated into SQL queries and executed against the PostgreSQL database, with results displayed via the user interface.

REFERENCES

[1] Dario Amodei, Rishita Anubhai, Eric Battenberg, Carl Case, Jared Casper, Bryan Catanzaro, Jingdong Chen, Mike Chrzanowski, Adam Coates, Greg Diamos, Erich Elsen, Jesse Engel, Linxi Fan, Christopher Fougner, Tony Han, Awni Hannun, Billy Jun, Patrick LeGresley, Libby Lin, Sharan Narang, Andrew Ng, Sherjil Ozair, Ryan Prenger, Jonathan Raiman, Sanjeev Satheesh, David Seetapun, Shubho Sengupta, Yi Wang, Zhiqian Wang, Chong Wang, Bo Xiao, Dani Yogatama, Jun Zhan, and Zhenyao Zhu. 2015. Deep Speech 2: End-to-End Speech Recognition in English and Mandarin. arXiv:1512.02595 [cs.CL]

[2] William Chan, Navdeep Jaitly, Quoc V. Le, and Oriol Vinyals. 2015. Listen, Attend and Spell. arXiv:1508.01211 [cs.CL]

[3] Yuan-Hua Ni, Xun Li, Ji-Feng Zhang, and Miroslav Krstic. 2018. Equilibrium Solutions of Multi-Period Mean-Variance Portfolio Selection. arXiv:1803.08500 [math.OC]

[4] Yuanfeng Song, Raymond Chi-Wing Wong, Xuefang Zhao, and Di Jiang. 2022. Speech-to-SQL: Towards Speech-driven SQL Query Generation From Natural Language Question. arXiv:2201.01209 [cs.DB]

[5] Xuefang Zhao Di Jiang Yuanfeng Song, Raymond Chi-Wing Wong. 2012. VoiceQuerySystem: A Voice-driven Database Querying System Using Natural Language Questions. (2012). https://dl.acm.org/doi/10.1145/3514221.3520158#sec-cit

[6] Qian Zhang, Han Lu, Hasim Sak, Anshuman Tripathi, Erik McDermott, Stephen Koo, and Shankar Kumar. 2020. Transformer Transducer: A Streamable Speech Recognition Model with Transformer Encoders and RNN-T Loss. arXiv:2002.02562 [eess.AS]

[7] Victor Zhong, Caiming Xiong, and Richard Socher. 2017. Seq2SQL: Generating Structured Queries from Natural Language using Reinforcement Learning. arXiv:1709.00103 [cs.CL]