International Journal of Enhanced Research in Science, Technology & Engineering
ISSN: 2319-7463, Vol. 15 Issue 3, March-2026

Unified Data Communication System with Voice-Driven Access

Palak Agrawal[1], Gayatri Pillai[2], Sakshi Marbhal[3], Prajwal Padole[4], Dr. Girish Potdar[5]

[1,2,3,4,5]Department of Computer Engineering, Pune Institute of Computer Technology, Pune, India

ABSTRACT

In modern enterprises, data is fragmented across structured databases (SQL/NoSQL) and unstructured repositories such as documents, emails, and reports, creating barriers to unified information access and decision-making. This paper introduces IntelliQuery, an intelligent data communication framework that allows natural language and voice-enabled querying across structured and unstructured data sources. The proposed framework introduces a hybrid query engine that dynamically generates SQL queries for structured data and retrieves unstructured data using semantic vectors to merge into a unified ranking schema. A communication graph, modeled as a knowledge graph, maintains inter-entity relationships across heterogeneous datasets, supporting context-aware, multi-turn conversations and relationship-driven query expansion. A voice-enabled interface complements the framework using speech-to-text and text-to-speech modules to enhance accessibility and usability for non-technical users. The framework maintains SQL transparency, allowing users to view or modify any generated SQL code for interpretability and trust. Experimental evaluations demonstrate im-proved ease and efficiency of data retrieval while reducing response time compared to traditional search and BI tools. The combination of hybrid querying, context-driven knowledge graphs, and voice interaction offers a novel approach to intelligent and accessible data communication.

Keywords: Voice-based interaction, Retrieval-Augmented Generation (RAG), Large Language Models (LLM), Document Summarization, Speech Recognition, Natural Language Processing (NLP), Automatic Speech Recognition (ASR)

INTRODUCTION

Databases form the backbone of today's digital world, but interacting with them is not always straightforward. Organizations across various industries such as finance, healthcare, education and government generate and store vast amounts of information in a range of formats. The structured data is stored in relational databases (SQL/NoSQL) while an equally large portion of important information is stored in unstructured sources such as documents, reports, and emails. Extracting meaningful insights from these dissimilar systems often requires multiple tools and consider-able technical expertise, creating barriers for non-technical users and slowing decision-making. Non-technical users face challenges for retrieving information without the assistance of data engineers which may lead to slower decision making cycles and reduced operational efficiency.

Conventional Business Intelligence (BI) tools and search engines attempt to address these difficulties but can handle only one data type, either structured or unstructured. This results in fragmented and incomplete analysis. However, they heavily depend on predefined queries, dashboards, and schema knowledge , which restricts their ability to adapt to dynamic user requirements. This limitation prevents organizations from realizing the potential of their data ecosystems. This gap translates into accessibility and usability issues from the user's perspective.

With the recent advancements in Natural Language Processing (NLP) and Large Language Models (LLMs), intelligent and context-aware querying can bridge the gap between structured and unstructured data. These technologies enable systems to interpret human-like language and intent, allowing users to ask questions, in text or voice, with-out prior knowledge of the database schema. By understanding the semantics of user queries and mapping them to structured and unstructured data, AI-driven systems can deliver comprehensive and context-aware results with minimal latency.

This paper proposes **IntelliQuery**, a system that integrates NLP-based query understanding, semantic vector search, and SQL generation in one unified platform, enabling users to query structured and unstructured data through natural language or voice. The framework includes a hybrid query engine, knowledge graph for contextual understanding, and a transparent SQL view for user trust and explainability.

LITERATURE SURVEY

The evolution of natural language based database interaction has undergone a fast paced transformation over the years, driven by the advancements in Natural Language Processing (NLP), Automatic Speech Recognition (ASR), and deep learning. Initially research mainly focused on converting text to SQL queries through handcrafted linguistic rules, while the recent approaches have shifted to neural and end-to-end models capable of understanding complex queries.

The conceptual framework to SQL query formation for databases using NLP (Aditya narhe et al. (2019))[2] lever-aged voice recognition and text processing, employing logistic regression for query classification and SQL formation. The study demonstrates the possibility of translating natural language into SQL commands, marking one of the earliest efforts to make the databases more accessible to users.

In the same year, Jiaqi Guo et al.(2019) [5] introduced IRNet, a model that translates complex natural language queries to SQL through an intermediate representation called SemQL. This representation was a turning point, enabling the handling complex joins, nested conditions, and multi-table queries across various database schemas. IRNet achieves better generalization and cross domain adaptability by isolating the linguistic understanding process from the constraints of database syntax.

Advancing accessibility, a system was developed regex-based recognition[3] to process CRUD (Create, Read, Up-date, Delete) operations from verbal inputs. Providing more emphasis on assistive technology, TalkSQl specifically targeted for visually impaired users. It allows them to interact with databases through speech, a remarkable step towards inclusive data systems.

Around the same time, Pooja Choudhari, Rutuja Kadam, and Namrata Kamble (2020)[6] proposed an approach combining the voice authentication with NLP-driven query generation to ensure data security along with accessibility. This integration of user verification with query generation introduced the concept of secure voice-based querying, an essential consideration for enterprise adoption.

With the evolution of deep learning architecture, researchers began to move beyond modular pipelines towards end-to-end neural models[1]. This work mainly focuses on SpeechSQLNet, a neural architecture capable of translating the speech directly to SQL without intermediate text conversion. This model integrates speech and schema encoders to achieve contextual understanding, significantly reducing dependency on manual preprocessing and improving real-time query generation accuracy.

Further extending this direction, Song et al.(2022)[7] proposed a system that directly mapped raw speech inputs to SQL outputs, creating a fully integrated speech-to-query pipeline. By eliminating the intermediate transcription, this methodology enhanced both speed and reliability, setting a new benchmark for real-time voice-based database interaction.

In the recent years, a system focusing on post-processing ASR outputs[4] was used to improve SQL readability and coherence. This study did not propose a new model architecture, but addressed a crucial weakness in earlier systems, i.e. the degradation of query accuracy due to speech recognition errors and hence proposed a linguistic refinement as a corrective mechanism.

From rule-based grammars to deep nerual architectures, the research in voice-driven SQL query generation has made a substantial progress towards natural and accessible database interaction. The existing systems had several shortcomings. They mainly focused on structured data, overlooking the massive volume of unstructured information in the form of text documents, reports, and emails that play a vital role in organizational knowledge. While some systems offer end-to-end speech-to-SQL translation, they lack context awareness, multi-turn conversational ability, and integration with modern Large Language Models (LLMs) capable of semantic reasoning. It had lack of transparency which made users incapable of easy interpretation or modification of the generated SQL query. This further diminishes trust and explainability.

To overcome these gaps, the proposed framework of IntelliQuery introduces a unified, hybrid querying model capable of integrating structured and unstructured data access, context-driven knowledge graphs, and voice-enabled natural language interfaces, moving a step closer to enhancing data communication.

METHODOLOGY

Our proposed system IntelliQuery enables conversation with structutred and unstructured data. It allows users, whether they're a technical expert or not, to ask questions about data in plain English, either by typing or speaking in their natural language. When users speak, the system uses powerful speech recognition tool of ASR i.e Google's Speech-to-Text to turn their voice into text. From there, IntelliQuery's smart language models, such as BERT, SBERT, spaCy, and NLTK — analyze the text to detect the intent and understand exactly what the user means, identify key terms and entities, and figure out what kind of data the user is looking for.

Our system is flexible and can handle fetching data from both structured and unstructured databases. It follows modular design, so each part works smoothly with the others. Once it understands the user's question and intent, IntelliQuery decides whether the query is about structured data (like rows and columns in a database), unstructured data (like documents, pdfs, reports, or text files), or a mix of both. If it's structured, the system automatically converts the question into an optimized SQL query to fetch the right information from structured database. If it's unstructured, it uses modern search tools such as FAISS to find relevant information based on the semantics and not just keywords. When a query involves both types of data, IntelliQuery combines the results intelligently to give one clear and complete answer.

One of the most important thing about IntelliQuery is, its ability to hold multi-turn conversations. This means it remembers what was asked earlier, understands follow-up questions if there is any lack of information, and connects across different data sources. It uses a relationship-aware communication graph to link related information and keep track of context.

Finally, IntelliQuery displays output of the user's query and makes information easy to understand and visualize. The result is displayed along with the knowledge graph, so that users can see the links and relationships visually. For users who want the output in an audible form, IntelliQuery can read the results aloud using Text-to-Speech (TTS).

IntelliQuery uses authenticated access and role-based access control to ensure only authorized users can view sensitive data. It keeps the system safe, secure and enterprise-ready.

Hence, IntelliQuery changes how people interact with data. It removes the need to know coding or query languages, allowing everyone to explore information naturally — just by asking questions through natural conversation.

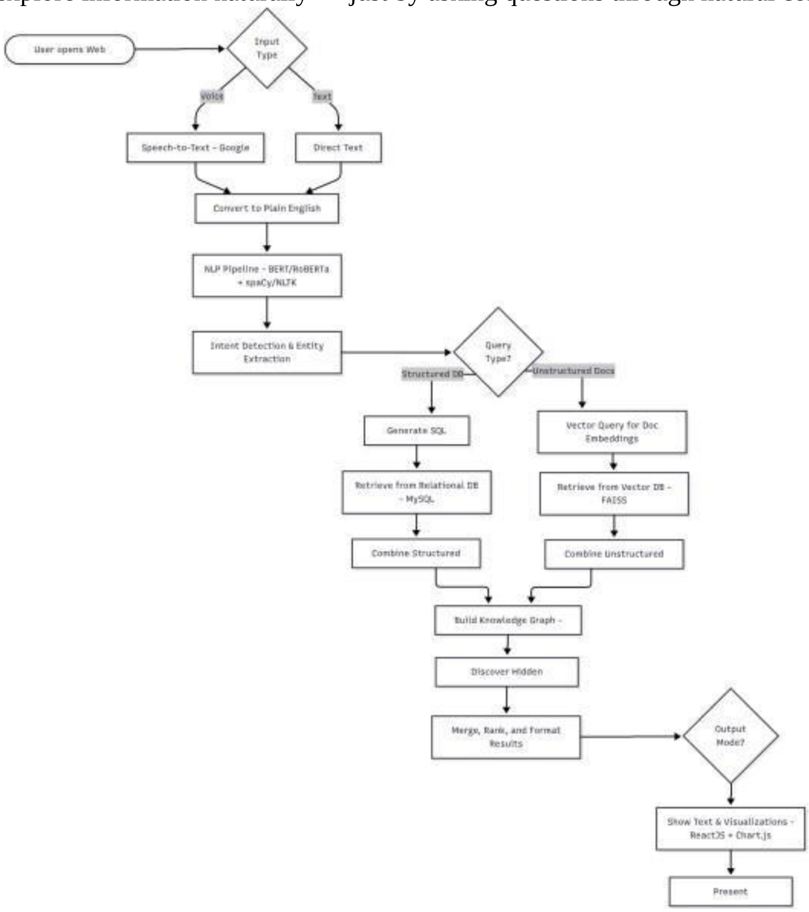

Figure 1: Flow Diagram of IntelliQuery

SYSTEM ARCHITECTURE

The IntelliQuery framework is built upon a modular, multi-tier architecture designed to facilitate seamless natural language interaction across heterogeneous data environments. The process is initiated at the Presentation Layer, where the system accepts multimodal inputs in either textual or acoustic form. For vocal queries, the architecture integrates an

Automated Speech Recognition (ASR) module—specifically Google Speech-to-Text—to transcribe spo-ken data into a standardized textual format. This ensures downstream processing remains agnostic to the initial input modality. Once digitized, the query enters the Intelligence Layer, where a robust Natural Language Process-ing (NLP) pipeline utilizes transformer-based models (such as BERT or RoBERTa) and linguistic libraries (spaCy, NLTK) to perform intent detection, entity extraction, and query classification. This stage is critical for disam-biguating user objectives and identifying whether the request necessitates retrieval from structured, unstructured, or hybrid data repositories.

The core computational efficiency of the system resides in the Hybrid Query Engine. For requests targeting structured data, the engine autonomously translates natural language into optimized SQL statements for execution against relational databases like MySQL. Conversely, for unstructured data—such as reports, emails, or PDFs—the system employs Semantic Vector Search. Using FAISS (Facebook AI Similarity Search), the engine generates high-dimensional embeddings to retrieve information based on semantic relevance rather than simple keyword matching. In hybrid scenarios, a unified ranking schema merges results from both domains to ensure contextual alignment.

To enhance the depth of the retrieved information, the architecture incorporates a Communication Graph. This specialized knowledge-graph layer maps inter-entity relationships and dependencies extracted from the data sources, facilitating multi-turn conversational interactions and relationship-driven query expansion. This graph-based ap-proach transforms raw data into a structured knowledge network, which is subsequently rendered through an interac-tive interface. To ensure accessibility and a closed-loop user experience, the system concludes with a Text-to-Speech (TTS) module, vocalizing synthesized insights for the user. By integrating ASR, neural-symbolic NLP, and graph- driven retrieval, the architecture provides a robust, context-aware solution for complex data communication.
Figure 2 illustrates the high-level architecture of IntelliQuery, showing the flow from voice/text input to final result presentation.

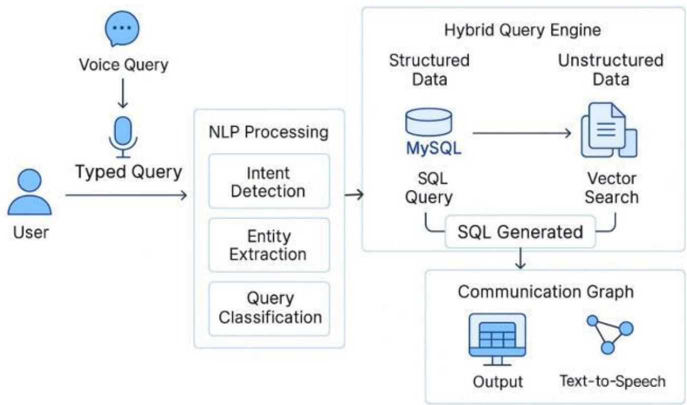

Figure 2: High-level architecture of the IntelliQuery system.

MODULE DESIGN

The proposed framework can be broadly grouped in seven modules: voice input module, sppech-to-text module, NLP processing module, hybrid query module, communication graph, text-to-speech module and system integration and API layer. The purpose of each of the modules is explained in the following part.

i. Voice Input Module:

This module captures the user's voice query through a microphone using our web interface. The audio recorder picks voice of an user, reduces noise and cleans up background sounds. It ensures that the audio is clean and ready for speech recognition.

ii. Speech-to-text Module

It converts spoken voice input into readable text in plain English using Google Speech-to-Text APIs. The Recorder audio is sent to STT engine and it is converted into text using real-time transcription. The built-in error handling corrects minor inaccuracies in speech. This module gives an accurate representation of voice query.

iii. NLP Processing Module

This modules aims to understand the meaning behind user's query. It consist of 3 submodules: Intent Detection - it figures out what the user wants
Entity Extraction - it chooses key details like region, entity, date etc.
Query Classification - it determines if the query targets structured data or unstructured data.

iv. Hybrid Query Engine:

It serves the purpose of execution of the query on the data. The SQL Processor runs SQL queries on structured databases. Vector Search Engine uses embeddings for semantic search on unstructured text such as documents and pdfs. Finally, the result aggregator combines results from both and sends a unified and relevant response as an output.

v. Communication Graph Module:

This graph maintains inter-entity relationships across heterogeneous data resources, which supports context-aware, multi-turn conversations and relationship-driven query expansion. The result displayed along with knowledge graph helps in visualizing the links and relationships between entities.

vi. Text-to-Speech Module

It Converts the text output into natural language speech using tools like Google TTS and AWS Polly & provides a spoken response that completes the voice-based interaction.

vii. System Integration & API Layer

The last module of our system which connects all other modules and ensures smooth end-to-end operation of each microservice. REST APIs are used for communication. Authentication is also implemented for secure data access.

APPLICATIONS

Proposed framework fits into many applications as in enterprise and business intelligence, legal document analysis, etc.

i. Enterprise and Business Intelligence:

Voice-based Business Analytics Dashboards: Managers can ask queries like "Show me quarterly revenue by region" and receive instant visual dashboards.

Data-Driven Decision Support: Helps non-technical executives interact with company databases using plain English or voice queries.

ii. Academic and Research Assistance :

Students and researchers can use IntelliQuery to retrieve insights from vast collections of research papers, thesis and reports using natural language queries. Instead of manually searching through the databases, they could receive summarized, citation-supported results. this accelerates the knowledge discovery and encourages exploratory research.

Intelligent Research Assistants: Students or researchers can query research databases, libraries, or repositories using voice commands.

Campus Data Portals: University administrators can ask questions like "List students with pending fee pay-ments" or "Show placement statistics".

iii. Corporate Knowledge Management – Intelligent document retrieval for organizations:

Organizations often struggle to retrieve relevant internal documents spread cross emails, reports and databases. the system enables employees to find precise information quickly by asking conversational questions, improving productivity and decision making. It also helps preserve institutional knowledge through intelligent , searchable archives.

iv. Environmental Policy Research – Analysis of complex environmental datasets:

Environmental researchers and policymakers can use the system to analyze structured and unstructured data sources. This holistic data access supports better understanding of climatic trends and facilitates evidence-based policymaking.

v. Healthcare and Medical Systems

Electronic Health Record (EHR) Access: Doctors can query patient data like "Show latest blood test results for John Doe" without manual typing.

Medical Research Retrieval: Quickly summarize clinical trial data or patient outcomes using natural queries. Hospital Management Systems: Voice-enabled dashboards for inventory, staff schedules, and patient admis-sions.

vi. Research and Data Science

Data Preprocessing via Voice: Automates tasks like "Clean missing values in sales dataset" or "Generate correlation matrix."

Exploratory Data Analysis (EDA): Natural language-driven insights generation for analysts.

Integration with LLM-powered Notebooks: Enables conversational querying of code outputs and datasets in Jupyter/Colab environments.

CONCLUSION AND FUTURE SCOPE

Voice-based Natural Language Querying (NLQ) systems have significantly evolved from traditional rule based frameworks to modern neural, end-to-end architectures. These advancements have enabled the users , both technical as well as non-technical, to interact with the data more efficiently and intuitively. This is done using speech and text, across structured and unstructured resources. Systems like IntelliQuery demonstrate that hybrid approaches, combining SQL translation for structured data and semantic vector search for unstructured data can provide seamless and unified access to diverse information, Moreover, incorporating multi-turn conversation handling and context-aware entity linking allows such systems to maintain coherent dialogues, remember references, and resolve complex queries effectively.

Despite these advancements, several challenges remain before voice-based NLQ can achieve real-worlds applicability. Schema generalizatin and code-mixed query support and adaptive reasoning across heterogeneous data resources are still active research areas. Addressing these challenges will not only improve system robustness but also move us closer to truly enhancing data access, enabling conversational interfaces that are intuitive, context-aware, and widely accessible across industries and user groups.

Future research may focus on:

i. **Integration with Multi-Modal LLMs :**
Incorporate vision-language models (like GPT-4V or Gemini) to allow querying images, charts, or scanned PDFs using voice or text.

ii. **Conversational Memory and Context Persistence :**
Extend the system to retain long-term conversational context and past session history using vector memory, enabling continuous, multi-session understanding.

iii. **Integration with Enterprise Tools :**
Connect IntelliQuery with tools like Jira, Notion, Salesforce, or Confluence for seamless enterprise data access and knowledge management.

iv. **On-device Edge Deployment :**
Develop lightweight models optimized for mobile or IoT devices, allowing offline or low-latency voice-based querying in industrial or rural environments.

v. **Enhanced Data Security and Privacy :**
Implement privacy-preserving federated learning and end-to-end encryption to enable secure, decentralized querying across distributed databases.

vi. **Real-time Streaming Data Integration :**
Extend capabilities to handle real-time data from APIs, IoT devices, and sensors, making the system applicable to fields like smart cities or environmental monitoring.

vii. **Multilingual and Code-Mixed Query Expansion :**
Support regional languages and code-mixed speech (like Hindi-English), leveraging multilingual NLP models for accessibility in India and global markets.
Addressing these challenges will move us closer to democratizing data access through truly conversational systems.

REFERENCES

[1]. Yuanfeng Song, Raymond Chi-Wing Wong, Xuefang Zhao, and Di Jiang. "VoiceQuerySystem: A Voice-driven Database Querying System Using Natural Language Questions." SIGMOD, 2022. DOI: 10.1145/3514221.3520158.
[2]. A. Narhe, C. Mohite, R. Kashid, P. Tade, and S. Waghmode. "SQL Query Formation for Database System using NLP." IJERT, 2019.
[3]. G. Obaido, A. Ade-Ibijola, and H. Vadapalli. "TalkSQL: A Tool for the Synthesis of SQL Queries from Verbal Specifications." IMITEC, 2020.
[4]. M. Suresh Kumar, K. Jayasundar, and S. Ragavendhiran. "Propagation of SQL by ASR Using NLP." ICCEBS, 2023.
[5]. Jiaqi Guo et al. "Towards Complex Text-to-SQL in Cross-Domain Databases with Intermediate Representation." ACL, 2019.
[6]. "Implementation on Voice-based SQL Query Generation using NLP." JETIR, 2020.
[7]. Yuanfeng Song et al. "Speech-to-SQL: Towards Speech-driven SQL Query Generation from Natural Language Questions." VLDB Journal, 2024.

[8]. Huadai Liu et al. "Wav2SQL: Direct Generalizable Speech-To-SQL Parsing." Findings of ACL, 2024.
[9]. Ahmed Elgohary et al. "Speak to Your Parser: Interactive Text-to-SQL with Natural Language Feedback." ACL, 2020.
[10]. Arun Kumar et al. "SpeakQL: Towards Speech-driven Multimodal Querying." CHI, 2021.
[11]. Aiwei Liu et al. "A Comprehensive Evaluation of ChatGPT's Zero-Shot Text-to-SQL Capability." arXiv, 2023.
[12]. Jing Chen, Yu Zhang, and Jiaheng Lu. "Text-to-SQL Meets the Real-World." ICEIS, 2024.
[13]. Yifan Wu, Shuai Zhang, and Lei Chen. "A Survey on Deep Learning Approaches for Text-to-SQL." VLDB Journal, 2022.
[14]. Hongyi Wang, Zheng Li, and Wenjie Zhang. "Large Language Model Enhanced Text-to-SQL Generation: A Survey." IJISA, 2024.
[15]. Yuanfeng Song et al. "SpeechSQLNet: Toward End-to-End Speech-driven SQL Query Generation." VLDB Journal, 2024.
[16]. Xu, Xiaojun & Liu, Chang & Song, Dawn. (2017). "SQLNet: Generating Structured Queries From Natural Language Without Reinforcement Learning." 10.48550/arXiv.1711.04436..
[17]. Ziwei Bai, Bo yu, Bowen Wu, Zhuoran Wang, Baoxun Wang "Learning to generate structured queries from natural language with indirect supervision, Computer Speech & Language ." Volume 67, 2021, 101185, ISSN 0885-2308.
[18]. Nascimento E., Garc´ıa G., Feij´o L., Victorio W., Izquierdo Y., R. de Oliveira A., Coelho G., Lemos M., Garcia R., Leme L. and Casanova M. (2024). "Text-to-SQL Meets the Real-World ." In Proceedings of the 26th International Conference on Enterprise Information Systems - Volume 1: ICEIS; ISBN 978-989-758-692-7.
[19]. V. Zhong and C. Xiong and R. Socher "Seq2sql: Generating structured queries from natural language using reinforcement learning." arXiv preprint arXiv:1709.00103, (2017)..
[20]. I. Androutsopoulos, G. D. Ritchie and P. Thanisch "Natural language interfaces to databases–an introduction," Natural language engineering, vol 1, pp 29–81, (1995).
[21]. O. Popescu, N. P. A. Vo, V. Sheinin, E. Khorashani and H. Yeo "Tackling complex queries to relational databases," Intelligent Information and Database Systems: 11th Asian Conference, ACIIDS 2019, Yogyakarta, Indonesia, April 8–11, 2019.
[22]. H. Li and Y. Shi . "A wordnet-based natural language interface to relational databases," 2nd International Conference on Computer and Automation Engineering (ICCAE) Vol. 1, pp. 514–518, IEEE, (2010) .