AI-Powered Data Interaction: A Natural Language Chatbot for CSV, Excel, and SQL Files

Dhanya K.R.[1], Venkatesh S.[2]

[1,2] Computer Science with Data Analytics, Dr N.G.P Arts and Science College, Coimbatore, India

E-mail: [1]dhanyafab@gmail.com, [2]venkateshkaviya1234@gmail.com

Abstract

Artificial Intelligence (AI) is revolutionizing data interaction by making it more efficient, accessible, and user-friendly. Traditionally, extracting insights from structured data stored in CSV files, Excel spreadsheets, and databases required manual processing through SQL queries and mathematical formulas. These conventional methods were not only time-consuming and prone to human error but also demanded significant technical expertise and also limiting accessibility for the non-technical users. To overcome these challenges, an AI-powered Data Agent has been developed to automate data interaction through natural language queries. By utilizing advanced Large Language Models (LLMs) such as Google Gemini 1.5 Pro, in combination with frameworks like Streamlit, LangChain, and Pandas, the system processes the structured data and retrieves relevant insights. Unlike traditional methods that return raw database records, this system generates responses in a conversational Q&A format, making complex data more comprehensible and actionable. This approach significantly reduces the manual effort involved in data extraction, minimizes the risk of errors, and enhances decision-making efficiency. Business analysts, researchers, students, and organizations can benefit from this solution. Moreover, the integration of Google Gemini provides a cost-effective alternative to expensive data analysis tools, enabling seamless automation and democratizing access to data-driven insights. By bridging the gap between raw

DOI: https://doi.org/10.36548/jitdw.2025.1.001
Received: 27-01-2025. received in revised form: 21-02-2025, accepted: 06-03-2025, published: 18-03-2025

data and meaningful conclusions, this AI-based solution enhances productivity and enables informed decision-making across various domains.

Keywords: Artificial Intelligence, Data Interaction, Natural Language Queries, Large Language Models, Google Gemini, Streamlit, LangChain, Automation, Conversational AI.

1. Introduction

Data retrieval is the process of accessing, extracting, and presenting information from stored sources like CSV files, Excel spreadsheets, and SQL databases. Traditionally, retrieving data from these structured formats required technical expertise and manual effort [2]. In CSV files, data is stored in the form of plain text with values separated by commas, and retrieval methods range from basic manual searches in spreadsheet software to advanced filtering using programming languages like Python with Pandas [8]. Excel spreadsheets offer built-in functions like VLOOKUP, PivotTables, and VBA macros for data extraction, while SQL databases use structured queries (SELECT, JOIN, GROUP BY) to retrieve and manipulate data efficiently. However, these conventional approaches require knowledge of query syntax, formulas, or scripting, making data access difficult for non-technical users. Artificial Intelligence (AI), particularly the Large Language Models (LLMs) like Google Gemini, GPT-4, Claude, and Llama, is revolutionizing data retrieval by enabling users to interact with structured data using natural language queries[9-12]. These models eliminate the need for complex queries, making data access more intuitive and efficient. This research presents an AI-powered data retrieval system that automates the process by integrating Google Gemini 1.5 Pro with tools like Streamlit, LangChain. The system allows users to query structured files, such as CSV, Excel, and SQL databases, through conversational input and provides responses in a structured Q&A format rather than raw database records. This solution minimizes the manual effort, reduces errors, and accelerates the decision-making by making data retrieval more accessible to business analysts, students, and organizations. While designed for static datasets rather than real-time streaming data, this cost-effective AI-driven approach bridges the gap between raw data and meaningful insights, transforming the way structured data is accessed and analyzed[13-16].

2. Related Work

Several studies have referred to the use of Natural Language Processing (NLP) for data querying, particularly in enabling users to retrieve information from structured data formats such as SQL databases, CSV files, and Excel (XLSX) files. While much of the existing work focuses primarily on SQL-based systems, these studies served as inspiration to extend AI-driven data querying beyond SQL to support other structured file formats like CSV and Excel, thereby broadening the scope of data accessibility .Baig et al.[1] provides a comprehensive review of Natural Language to SQL (NL2SQL) systems, discussing various techniques for converting human language into structured queries and highlighting key challenges in handling complex queries. This work laid the foundation for further exploration of how natural language can interface with structured query languages. Katsogiannis-Meimarakis and Koutrika et al [3] examines deep learning approaches for text-to-SQL tasks, emphasizing the role of Large Language Models (LLMs) in improving query generation accuracy. Shaikh et al. [4] proposed an NLP-based system that utilizes machine learning techniques to optimize query structure, making data retrieval more accessible for non-technical users. Although these studies primarily focus on SQL databases, the present work extends the principles of NLP-driven querying to support not only SQL but also CSV and Excel files. This advancement bridges the gap between NLP-based database querying and broader data accessibility, enabling the systems to process and retrieve information from various file formats effectively. The idea for this approach emerged from recognizing the limitations of existing SQL-centric solutions and the need to make structured data more accessible across various formats [5-7].

3. Methodology

The methodology section outlines the approach and techniques used to develop the AI-driven data retrieval system that processes natural language queries and generates conversational responses based on structured data from CSV, Excel, and SQL database files. Mainly, it highlights the integration of large language models (LLMs) for query interpretation, ensuring an interactive and user-friendly way to extract insights from tabular data.

3.1 Data Collection and Processing

The sample dataset used in this research was manually created for demonstration purposes, although the developed system is designed to support large datasets as well. This

dataset consists of 10 student records containing details such as Name, Tamil, English, Maths, Science, Social Science, Total, and Rank. It was saved in multiple formats, including CSV, Excel (XLSX), and SQL database (.db), to showcase the system's ability to process different structured data sources. The dataset is well-organized with no missing values. This structured format allows the AI model to understand and respond to user queries in a natural, conversational manner, demonstrating how the system simplifies data interaction using NLP-based querying. The dataset used in this study is shown in Table 1.

Table 1. Student Data

Name	Tamil	English	Maths	Science	Social Science	Total	Rank
Arun	37	55	81	99	96	368	2
Bala	87	65	78	49	61	340	7
Charan	42	92	93	92	100	419	1
Deepak	92	44	38	71	87	332	8
Elan	84	67	95	64	43	353	5
Fathima	54	81	87	44	84	350	6
Gokul	72	74	95	63	57	361	3
Harini	92	95	42	80	52	361	3
Irfan	74	92	91	36	63	356	4
Jaya	38	48	43	81	82	292	9

3.2 AI Model Selection

For this research, a Large Language Model (LLM) was required to process the user queries and provide meaningful responses. While OpenAI's GPT models, such as GPT-4, are widely used for natural language processing, they come with usage costs that limit accessibility for free applications. Similarly, other models like Anthropic's Claude and Meta's LLaMA offer advanced text generation and understanding but may have restrictions in terms of integration and dataset handling. In contrast, Google Gemini 1.5 Pro provides a more cost-effective and efficient solution, making it an ideal choice for this research. Gemini is used in natural language understanding, and it analyzes the structured data from CSV, Excel, and SQL files, enabling the data retrieval through conversational interactions. Its integration into applications offers several advantages, including real-time querying, ease of deployment with frameworks like Streamlit and LangChain, and support for multimodal data processing. Moreover, Gemini's cloud-based architecture ensures accessibility without requiring extensive computational resources, making it suitable for both small-scale and large-scale data processing tasks. By

using Gemini, this research demonstrates a practical and easy approach to AI-driven data interaction, enabling users to extract insights from structured datasets efficiently without the need for complex query writing.

3.3 System Architecture

The system is designed to help users ask questions about their data and get AI-generated answers. It starts with the user uploading a file through the Streamlit UI. The system supports CSV, Excel, and SQLite files. Once uploaded, the file is processed, the CSV and Excel files are turned into tables using Pandas, while SQLite files are connected to a database. After the data is processed, the user can type a question in natural language. This question is processed by LangChain, which helps the AI understand it properly. Then, the Google Gemini API analyzes the data and generates a meaningful response. The answer is displayed back in the Streamlit UI in a simple and clear format. This system allows users to easily explore their data using AI, with the help of Python, Pandas, Streamlit, LangChain, and Google Gemini API for smooth processing and accurate responses. Figure 1 shows the system architecture.

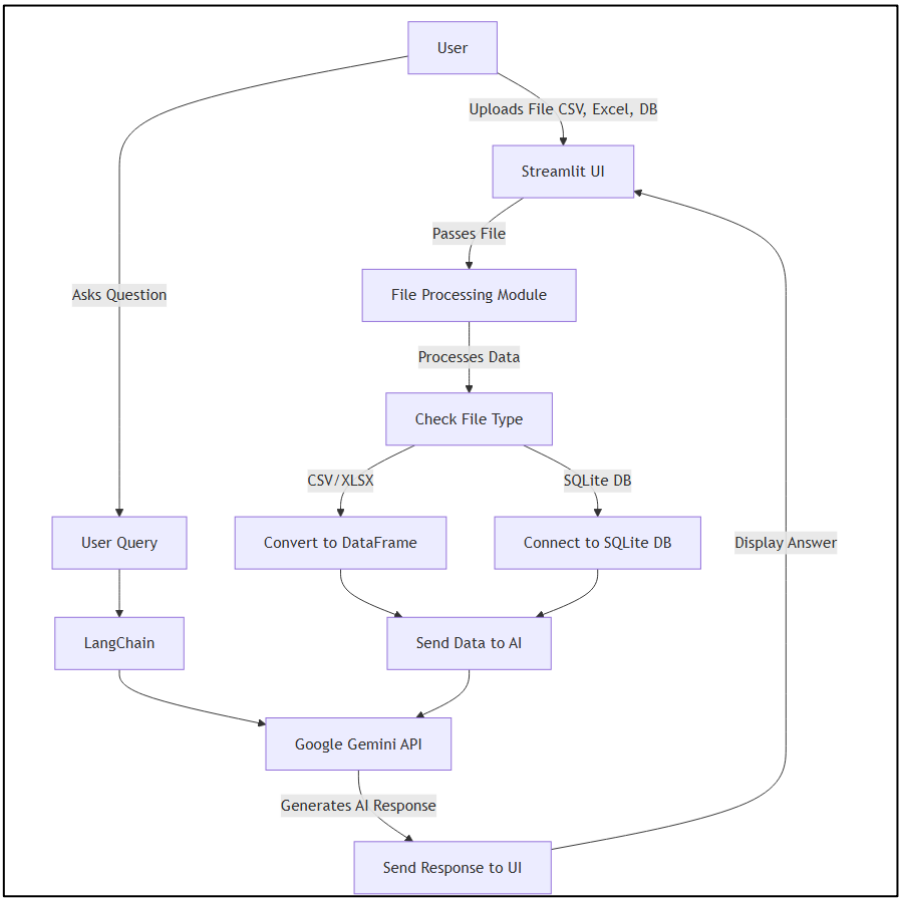


Figure 1. System Architecture

3.4 Natural Language Query Processing

Natural Language Query Processing refers to the process of interpreting and transforming user input in natural language into meaningful queries that can interact with structured data, such as CSV, Excel, or SQL database files. In the system, this begins when the user submits a question or request about the uploaded data, such as "What is the total score of the highest-ranked student?" or "Show me students with grades above 80 in Maths." The system first processes the natural language query using Google Gemini AI, which understands the context, intent, and relevant dataset details. It breaks down the query, recognizes keywords, and maps them to the structured data. Figure 2 shows the Prompt Used for Gemini AI.

"You are an AI assistant that helps users interact with structured data, such as CSV, Excel, and SQL databases, using natural language. When a user asks a question about their dataset, analyze the query, understand its intent, and generate an appropriate structured query in SQL (for databases) or Pandas (for CSV/Excel). Ensure accuracy by considering column names, values, and possible ambiguities. If necessary, make logical assumptions or ask for clarification. Return the results in a human-readable format, making the response easy to interpret."

Figure 2. Prompt used for Gemini AI

To ensure accuracy, Gemini also addresses the ambiguities by asking clarifying questions or making logical assumptions. For instance, if a user asks, "Show me top students," the AI might infer that they are referring to students with the highest total score and return the relevant data. Once the query is processed, it is converted into a structured SQL query (for databases) or a Pandas command (for CSV/Excel) for data retrieval.

For example: If a user asks, "What is the total score of the highest-ranked student?"

SQL Query: "SELECT Total FROM students ORDER BY Rank ASC LIMIT 1;

Pandas Command: "df.loc[df['Rank'].idxmin(), 'Total']

If a user asks, "Show me students with grades above 80 in Maths,

SQL Query: "SELECT * FROM students WHERE Maths > 80;

Pandas Command: "df[df['Maths'] > 80]"

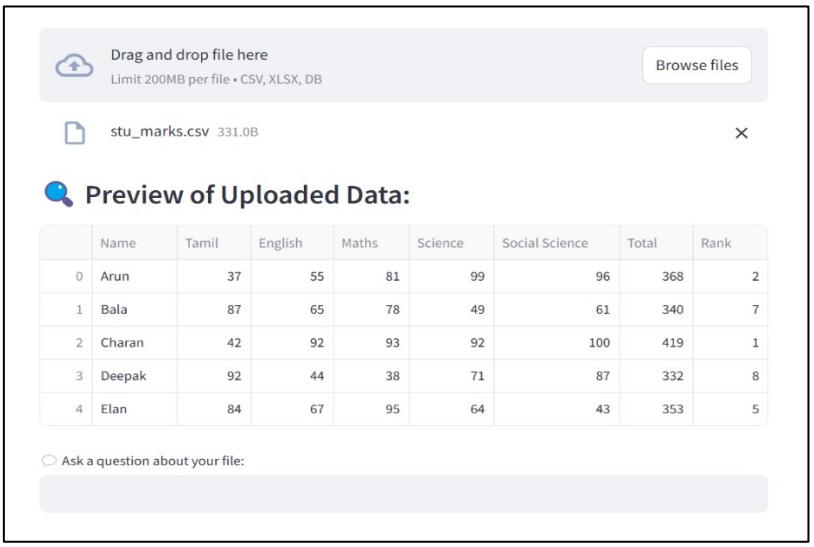
The final output is displayed in the Streamlit UI, formatted in a human-readable manner, ensuring that users can access insights without needing to understand SQL or Python.

4. Result and Discussion

This section deals with the results and discussion of the AI Data Chatbot, which supports CSV, Excel, and SQLite files.

Figure 3. Streamlit Interface

As shown in Figure 3, The interface, built using Streamlit in Python, offers an easy platform for uploading files through a drag-and-drop feature or the "Browse Files" button. After a file, such as stu_marks.csv, is uploaded, the system displays a preview of the dataset in a table, as shown in Figure 4. Below the table, an input field labelled "Ask a question about your file" allows for natural language queries. The AI model processes these questions, interprets the data, and generates appropriate responses.



	Name	Tamil	English	Maths	Science	Social Science	Total	Rank
0	Arun	37	55	81	99	96	368	2
1	Bala	87	65	78	49	61	340	7
2	Charan	42	92	93	92	100	419	1
3	Deepak	92	44	38	71	87	332	8
4	Elan	84	67	95	64	43	353	5

Figure 4. User Uploads a CSV File

A notable strength of the system is its ability to handle ambiguous or incomplete queries. For instance, when the query "Show me top students" was asked, the system understands that "top students" referred to those with the highest total scores and provided the expected results. Figure 5 demonstrates how the chatbot handles queries related to CSV files.

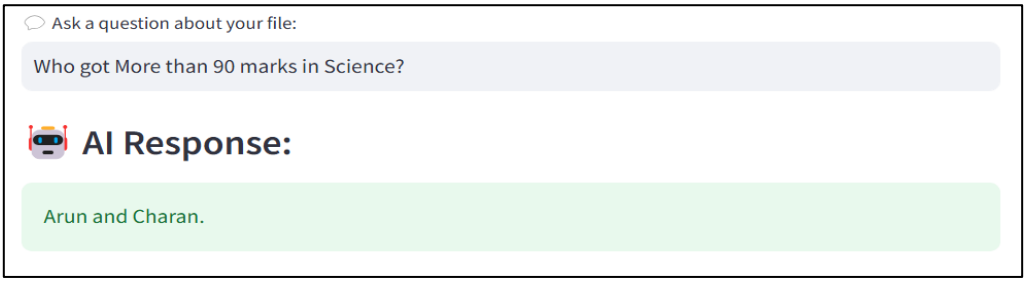


Figure 5. AI Response

In cases where queries were missing details, the system either made reasonable guesses based on the dataset structure or prompted for clarification, improving accuracy. Table 2 compares user queries with the system's responses, showcasing the system's ability to adapt to various data-related questions.

Table 2. Tested Queries and their AI responses

User Queries	AI Response
1. Who got First Rank?	Charan got the first rank

2. Who got Top 3 marks?	Charan (419), Arun (368), Harini (361)
3. What is the Rank of Jaya?	Jaya got the 9th Rank
4. What is Harini's Tamil Mark?	Harini got 92 in Tamil

This AI-powered approach enhances data accessibility, allowing non-technical users to extract meaningful insights without needing SQL knowledge. The conversational interface simplifies data retrieval, reducing the time and effort required for analysis. While the system performed well in most cases, some limitations were observed. The chatbot occasionally struggled with complex, multi-part queries that required advanced reasoning beyond basic data retrieval. Additionally, response times varied depending on the dataset size and query complexity, suggesting areas for optimization. Overall, the results demonstrate that the AI-powered chatbot processes natural language queries, retrieves accurate information, and improves user interaction with structured data. Future enhancements could focus on refining the query interpretation mechanism, handling multi-step queries more effectively, and optimizing performance for larger datasets.

5. Conclusion

In conclusion, the AI Data Chatbot makes it easy for users to ask questions and get answers from CSV, Excel, and SQLite files without needing to know SQL or programming. It uses Streamlit, LangChain, and Google Gemini API to process queries and clearly show results. This chatbot works well for medium-sized datasets, but it is not designed for live or streaming data. In the future, it can be improved to handle larger datasets and real-time data updates.

References

[1] Baig, Muhammad Shahzaib, Azhar Imran, Aman Ullah Yasin, Abdul Haleem Butt, and Muhammad Imran Khan. "Natural Language to SQL Queries: A Review." International Journal of Innovations in Science & Technology 4, no. 1 (2022): 147-162.

[2] Prasad, Akshar, Sourabh S. Badhya, Yashwanth YS, Shetty Rohan, Shobha G., and Deepamala N. "Enhancement of Natural Language to SQL Query Conversion Using Machine Learning Techniques." International Journal of Advanced Computer Science and Applications 11, no. 12 (2020): 494.

[3] Katsogiannis-Meimarakis, George, and Georgia Koutrika. "A Survey on Deep Learning Approaches for Text-to-SQL." The VLDB Journal 32 (2023): 905–936.

[4] Shaikh, Mohammed Osama, Fiza Shaikh, and Irfan Landge. "Natural Language to SQL Queries Generation Using NLP Techniques." International Journal of Advanced Research in Science, Communication and Technology 4, no. 7 (April 2024): 534.

[5] Narhe, Aditya, Chaitanya Mohite, Rushikesh Kashid, Pratik Tade, and Santosh Waghmode. "SQL Query Formation for Database System Using NLP." International Journal of Engineering Research & Technology 8, no. 12 (December 2019).1-13.

[6] Sonawane, Pankaj, Hetvi Shah, Sahil Doshi, and Aayush Parikh. "A Comprehensive Multimodal Analysis of Research Papers through Natural Language Processing (NLP) and Deep Learning." International Conference on Engineering, Science and Technology, International Journal of Scientific Research in Science, Engineering and Technology (IJSRSET), Volume 11, Issue 7, May-June-2024 No 1199-1206

[7] Behera, Santosh Kumar, and Mitali M. Nayak. "Natural Language Processing for Text and Speech Processing: A Review." International Journal of Advanced Research in Engineering and Technology 11, no. 11 (November 2020): 1947-1952.

[8] Meyer, Jesse G., Ryan J. Urbanowicz, Patrick C. N. Martin, Karen O'Connor, Ruowang Li, Pei-Chen Peng, Tifani J. Bright, Nicholas Tatonetti, Kyoung Jae Won, Graciela Gonzalez-Hernandez, and Jason H. Moore. "ChatGPT and Large Language Models in Academia: Opportunities and Challenges." BioData Mining 16 (2023): 20.

[9] Zhong, Victor, Caiming Xiong, and Richard Socher. "Seq2SQL: Generating Structured Queries from Natural Language Using Reinforcement Learning." arXiv preprint arXiv:1709.00103 (2017).

[10] Bora, Arunabh, and Heriberto Cuayáhuitl. "Systematic Analysis of Retrieval-Augmented Generation-Based LLMs for Medical Chatbot Applications." Machine Learning and Knowledge Extraction 6, no. 4 (2024): 2355-2374.

[11] Quidwai, Mujahid Ali, and Alessandro Lagana. "A RAG Chatbot for Precision Medicine of Multiple Myeloma." medRxiv (2024): 2024-03.

[12] Singh, Jaswinder. “Understanding Retrieval-Augmented Generation (RAG) Models in AI: A Deep Dive into the Fusion of Neural Networks and External Databases for Enhanced AI Performance.” Journal of Artificial Intelligence Research 2, no. 2 (2022): 258-275.

[13] "Retrieval-Augmented Generation Approach: Document Question Answering Using Large Language Model." International Journal of Advanced Computer Science and Applications (IJACSA) 15, no. 3 (2024).

[14] Kulkarni, Mandar, Praveen Tangarajan, Kyung Kim, and Anusua Trivedi. "Reinforcement Learning for Optimizing RAG for Domain Chatbots." arXiv preprint arXiv:2401.06800 (2024).

[15] Yu, Tao, Rui Zhang, Kai Yang, Michihiro Yasunaga, Dongxu Wang, Zifan Li, James Ma et al. "Spider: A large-scale human-labeled dataset for complex and cross-domain semantic parsing and text-to-sql task." arXiv preprint arXiv:1809.08887 (2018).

[16] Troy, Christopher, Sean Sturley, Jose M. Alcaraz-Calero, and Qi Wang. "Enabling generative ai to produce sql statements: A framework for the auto-generation of knowledge based on ebnf context-free grammars." IEEE Access 11 (2023): 123543-123564.