Natural Language to SQL Chain: A Dynamic Solution for SQL Query and Result Interpretation

Abhigyan Tejas Singh[1] Ashok Kumar Verma[2] Deepak Kumar[3] Himanshu Salal[4]
[1,2,3,4]Department of Computer Science and Engineering
[1,2,3,4]IIMT College of Engineering, Greater Noida, India

Abstract — This paper presents an innovative approach to bridging the gap between natural language and structured database querying by leveraging Google Generative AI (Gemini Pro) and advanced prompt engineering techniques. The proposed system enables users, regardless of technical proficiency, to input natural language questions that are automatically translated into SQL queries. These queries are executed on a relational database, and the results are subsequently transformed into easily comprehensible, human-readable responses. This approach addresses a critical need for intuitive data access and analysis, empowering stakeholders to retrieve insights without deep technical knowledge. Applications of this system span across business intelligence, customer support, and academic research, among others. The experimental results demonstrate the efficacy and accuracy of the proposed solution, emphasizing its potential to enhance user interaction with data systems.

Keywords: Natural Language to SQL, Generative AI, Google Gemini Pro, Prompt Engineering, Natural Language Interface (NLI), Query Translation, Relational Database Access, Human-Readable Output Generation, Intuitive Data Retrieval, AI-Powered Business Intelligence

I. Introduction

In the age of data-driven decision-making, the ability to access, query, and interpret structured data has become a fundamental requirement across industries. Relational databases serve as the backbone of modern information systems, storing vast amounts of structured data in a format optimized for retrieval through Structured Query Language (SQL). However, the technical nature of SQL poses a significant barrier for non-technical stakeholders who need to interact with databases to gain insights. This disconnects between users' natural language communication and the rigid syntax of SQL often necessitates the involvement of technical intermediaries, slowing down the decision-making process and increasing dependency on data specialists.

To address this challenge, recent advancements in Natural Language Processing (NLP) and Generative Artificial Intelligence (AI) offer promising solutions. In particular, large language models (LLMs) such as Google Gemini Pro have shown remarkable capabilities in understanding context, interpreting user intent, and generating coherent, task-specific outputs. These capabilities open the door to translating user queries expressed in natural language into precise, executable SQL statements. When paired with prompt engineering—an emerging technique for guiding LLM behavior through structured input prompts—the potential for intuitive, high-quality query generation becomes even more significant.

This paper presents an innovative system that combines Google Gemini Pro and advanced prompt engineering to automate the translation of natural language questions into SQL queries. The system is designed with accessibility and usability in mind, ensuring that users with little to no background in SQL or database management can retrieve data and insights seamlessly. Once a user inputs a question in plain English, the system processes the input through a carefully designed prompt, which guides Gemini Pro to generate the corresponding SQL query. This query is then executed on a connected relational database, and the results are formatted into a human-readable output, ensuring clarity and usability.

The implications of such a system are far-reaching. In business environments, decision-makers can gain real-time access to performance metrics without waiting for IT support. In customer service, support agents can extract relevant data quickly to resolve issues. In academic and research settings, users can interact with data repositories effortlessly, facilitating faster insights and discoveries. This democratization of data access empowers organizations to be more agile, efficient, and informed.

Moreover, the system's modular design allows for integration with various databases and customization for domain-specific applications. The use of prompt engineering ensures adaptability and fine-tuning, enabling the model to handle a wide range of query types, from simple data retrieval to complex aggregations and joins. Experimental results from real-world use cases demonstrate the system's accuracy, reliability, and ease of use, highlighting its value as a transformative tool in human-computer interaction with databases.

In conclusion, the fusion of generative AI, prompt engineering, and database technologies marks a pivotal step toward natural, intelligent, and accessible data interaction. This paper contributes to the growing body of research in AI-driven data systems by showcasing a practical, scalable solution to bridge the gap between natural language and structured data querying.

II. Literature survey

The task of translating natural language into SQL, commonly referred to as NL2SQL, has been a long-standing challenge in the intersection of natural language processing and database management. Early approaches to this problem relied heavily on rule-based systems and pattern matching, which were limited in scalability and adaptability to complex queries or domain-specific language. Over time, the rise of machine learning and deep learning models introduced data-driven methods that significantly improved accuracy and robustness.

Recent research has focused on neural architectures such as sequence-to-sequence (Seq2Seq) models, transformer-based models, and attention mechanisms to improve NL2SQL performance. Notable works like Seq2SQL and SQLNet introduced supervised learning techniques trained on large datasets such as WikiSQL, enabling more accurate SQL generation with better handling of query structures and database schema information. These models, however, often require extensive labeled data and retraining when adapting to new domains.

With the advent of large language models (LLMs) such as GPT-3, ChatGPT, and more recently Google Gemini Pro, researchers have explored few-shot and zero-shot learning for SQL generation. These models are capable of understanding and generating complex SQL queries based on minimal context, making them highly flexible. The introduction of prompt engineering has further enhanced LLM performance by structuring input prompts to guide the model's behavior effectively, reducing ambiguity in query interpretation.

Several recent studies have also emphasized the importance of explainable AI and human-centric design in NL2SQL systems to improve user trust and understanding. Integrating these systems into practical applications, such as business intelligence dashboards and chat-based data assistants, has demonstrated real-world viability and scalability.

This paper builds upon these advancements by leveraging Google Gemini Pro along with prompt engineering to create a practical, accurate, and user-friendly natural language interface for SQL query generation and execution.

III. METHODOLOGIES

The proposed system facilitates the translation of natural language questions into structured SQL queries using Google Gemini Pro, a powerful large language model (LLM), guided by advanced prompt engineering techniques. The objective is to bridge the gap between non-technical users and relational databases by enabling intuitive, accurate, and efficient data querying. The methodology comprises five key components: input handling, prompt engineering, SQL generation, query execution, and response interpretation.

A. Natural Language Input Handling

The user initiates the process by entering a query in natural language, such as "List all customers who placed orders in March." This input is captured via a user-friendly web interface or chatbot. To improve accessibility, the interface is designed to accommodate users with no technical background. Input sanitization and basic preprocessing (e.g., removal of extra spaces, correction of obvious typos) are applied to ensure clarity before processing.

B. Prompt Engineering

Prompt engineering is a crucial step in guiding the LLM toward generating accurate SQL output. A predefined template is used to construct the prompt, which includes:

- A brief description of the task.
- The structure of the relevant database schema (table names, columns, data types).
- A few examples of natural language queries and their corresponding SQL translations (few-shot learning).
- The user's input query.

This structured prompt provides Gemini Pro with enough context to understand the database domain and generate accurate queries. The prompt is updated dynamically based on the selected database and complexity of the query.

C. SQL Query Generation with Gemini Pro

The structured prompt is then passed to Gemini Pro through its API. Gemini Pro processes the prompt using its language understanding capabilities and returns an SQL query tailored to the user's request. The generated query is then parsed and checked for syntactic correctness using built-in parsers and validators.

D. Query Execution and Error Handling

Once the SQL query passes validation, it is executed on the connected relational database (e.g., MySQL, PostgreSQL, or SQLite). Execution is managed through a secure connection, with transaction management and logging to ensure traceability. Error handling mechanisms are implemented to manage cases where:

- The generated SQL contains syntax errors.
- The query references non-existent tables or columns.
- The database returns no results or unexpected results.

In such cases, the system either re-prompts Gemini Pro with refined context or provides meaningful feedback to the user, helping them refine their query.

E. Result Interpretation and Output Formatting

After successful query execution, the resulting data is processed and presented to the user in a clear, human-readable format. This may include tabular views, summary statements, or visualizations (e.g., charts or graphs), depending on the complexity of the result and user preference. This step enhances interpretability and helps non-technical users make sense of the data without needing to understand raw database outputs.

IV. FUTURE SCOPE

The integration of natural language processing (NLP) with structured data querying presents vast potential for future developments, especially as large language models (LLMs) like Google Gemini Pro continue to evolve. While the current system effectively bridges the gap between non-technical users and complex databases, several areas can be further enhanced to improve functionality, scalability, and user experience.

One significant direction for future work is the support for more complex SQL queries, such as nested queries, stored procedures, and real-time data aggregation. While current models handle basic SELECT, WHERE, and JOIN clauses effectively, real-world applications often demand multi-step logic, recursive queries, and advanced filters. Improving prompt design and training data can help LLMs understand and generate these advanced query structures more reliably.

Another area of growth lies in the integration of voice-based natural language inputs, allowing users to interact with the system using spoken commands. This would greatly benefit visually impaired users or professionals in fieldwork settings who need quick, hands-free access to data.

Additionally, multilingual support could be implemented to make the system globally accessible.

Currently, most LLMs are optimized for English, but adapting the platform to accept inputs in multiple languages will increase its usability across diverse user bases, especially in multilingual countries or organizations with international operations.

Security and access control mechanisms will also play a crucial role in future versions. As the system provides direct access to databases, ensuring that users can only query permitted data becomes critical. Role-based access, query auditing, and data anonymization are essential for enterprise deployment.

The system could also benefit from adaptive learning capabilities, where it learns from user interactions over time to refine query generation and response formatting. By incorporating user feedback loops, the model can improve accuracy and relevance in future interactions.

Finally, the future may see the integration of AI-generated data visualizations, enabling the model not only to return tabular data but also to automatically generate graphs, charts, and dashboards in response to user queries. This would turn the system into a powerful end-to-end analytics assistant.

In conclusion, as LLMs grow more powerful and adaptable, the system's scope can expand beyond simple question-to-query translation into a robust, intelligent platform for real-time data analysis, reporting, and decision support.

V. Results

The implementation of the proposed natural language to SQL query system using Google Gemini Pro and prompt engineering yielded promising results during experimental evaluation. A series of tests were conducted on a sample relational database containing sales, customer, and order information. The evaluation focused on the system's accuracy, efficiency, and usability across a variety of query types, including simple retrievals, aggregations, and conditional filters.

Out of 50 test queries written in natural language, the system successfully generated syntactically correct and semantically accurate SQL queries for 44 of them, resulting in an 88% success rate. These successful queries included examples such as "List all customers who made purchases in January," and "Show total sales by product category." The remaining 6 queries either produced incomplete SQL statements or misinterpreted the user's intent due to ambiguous phrasing. These errors were often resolved by refining the prompt or rephrasing the input query.

The response time was also satisfactory, with an average processing time of under 3 seconds per query, including prompt generation, model response, SQL execution, and result rendering. This demonstrates the system's viability for real-time applications in business intelligence or support environments.

Usability testing was conducted with a small group of non-technical users. Most participants reported that the system was intuitive and easy to use. They appreciated the ability to obtain database insights without needing SQL knowledge, and found the formatted results clear and useful.

Overall, the results indicate that the proposed system is not only feasible but also practical for real-world use. With minor improvements in handling edge cases and ambiguous language, the system has the potential to serve as a reliable natural language interface for relational databases.

VI. Discussion

The results of this study highlight the effectiveness and practicality of using large language models, specifically Google Gemini Pro, in bridging the gap between natural language queries and structured SQL queries. The system demonstrated high accuracy, responsiveness, and user-friendliness, supporting the central hypothesis that non-technical users can interact with relational databases through conversational input with minimal training or prior knowledge.

A key observation is the system's ability to interpret a wide range of natural language inputs accurately, especially when supported by well-structured prompt engineering. The success rate of 88% for correct SQL generation validates the strength of prompt design and the capabilities of Gemini Pro in understanding both natural language semantics and database schema context. This makes the system highly suitable for applications in business intelligence, customer support, education, and any domain where data access needs to be democratized.

However, some limitations were noted. The system struggled with highly ambiguous or complex queries involving nested logic, subqueries, or uncommon phrasing. These issues often resulted from insufficient schema context in the prompt or overly general instructions to the LLM. This indicates that while LLMs are powerful, their output quality is still sensitive to input structure and clarity. Enhancing the prompt dynamically based on query complexity or using context-aware chaining techniques could mitigate this issue.

Another discussion point involves scalability. While the system performed well on small to mid-sized databases, its performance in large-scale, real-time environments remains to be explored. Latency could increase with more complex queries or higher database loads, which may necessitate optimization strategies such as caching, asynchronous query processing, or hybrid model approaches.

From a usability perspective, users responded positively to the intuitive design and natural language interface. However, they also suggested integrating visual query builders and feedback mechanisms for learning and error correction, which could improve long-term engagement and system accuracy.

In conclusion, the discussion reveals both the strength and areas for improvement in the proposed system. With continued development, including more robust prompt designs, multilingual input, and adaptive learning features, this approach has the potential to transform how organizations and individuals interact with structured data.

VII. Conclusion

This research presents a novel approach to simplifying database interactions by translating natural language queries into structured SQL commands using Google Gemini Pro and advanced prompt engineering techniques. The system effectively bridges the gap between non-technical users and complex relational databases, enabling intuitive access to data insights without requiring programming or SQL expertise.

The results from experimental evaluation demonstrate the system's high accuracy, responsiveness, and user satisfaction. With an 88% success rate in generating correct SQL queries from natural language inputs, the solution proves to be reliable for various query types, including simple retrievals, conditional searches, and aggregate functions. Users also found the system easy to use and appreciated its ability to deliver results in a readable and meaningful format.

This work addresses a growing need in organizations for data-driven decision-making without requiring technical teams to handle every query request. By enabling business users, analysts, and support staff to interact directly with data using natural language, the system enhances productivity, reduces dependency on technical staff, and democratizes access to information.

However, the study also highlights areas for further improvement. Handling ambiguous or complex queries, improving response accuracy, and scaling the system for larger databases are essential next steps. Additionally, incorporating features such as voice input, multilingual support, and AI-powered data visualizations would significantly broaden the application scope.

In conclusion, the integration of generative AI with natural language interfaces represents a promising advancement in human-computer interaction for data analytics. As large language models continue to evolve, systems like the one proposed in this research will become increasingly powerful, accurate, and indispensable in both academic and business environments.

References

[1] Yaghmazadeh, N., Wang, Y., Dillig, I., & Dillig, T. (2017). Type- and Content-Driven Synthesis of SQL Queries from Natural Language. This paper introduces a technique that combines natural language processing, program synthesis, and automated program repair to synthesize SQL queries from natural language descriptions.

[2] Sun, Y., Tang, D., Duan, N., Ji, J., Cao, G., Feng, X., Qin, B., Liu, T., & Zhou, M. (2018). Semantic Parsing with Syntax- and Table-Aware SQL Generation. The authors present a generative model that maps natural language questions into SQL queries by considering the structure of tables and SQL syntax.

[3] Gan, Y., Chen, X., Xie, J., Purver, M., Woodward, J. R., Drake, J., & Zhang, Q. (2021). Natural SQL: Making SQL Easier to Infer from Natural Language Specifications. This study proposes an intermediate representation called Natural SQL (NatSQL) to simplify the translation from natural language to SQL.

[4] Cai, R., Xu, B., Yang, X., Zhang, Z., Li, Z., & Liang, Z. (2017). An Encoder-Decoder Framework Translating Natural Language to Database Queries. The paper discusses an encoder-decoder framework that translates natural language into SQL queries, integrating grammar rules to improve accuracy.

[5] Zhong, V., Xiong, C., & Socher, R. (2017). Seq2SQL: Generating Structured Queries from Natural Language using Reinforcement Learning. This research introduces Seq2SQL, a model that generates SQL queries from natural language using reinforcement learning techniques.

[6] Yu, T., Zhang, R., Yang, K., Yasunaga, M., Wang, D., Li, Z., Ma, Y., Yin, W., Lin, J., Li, S., & Radev, D. (2018). Spider: A Large-Scale Human-Labeled Dataset for Complex and Cross-Domain Semantic Parsing and Text-to-SQL Task. The authors present Spider, a dataset designed to evaluate the generalization ability of models in complex and cross-domain text-to-SQL tasks.

[7] Guo, J., Li, Z., & Zhang, X. (2019). Towards Complex Text-to-SQL in Cross-Domain Database with Intermediate Representation. This paper explores the use of intermediate representations to handle complex text-to-SQL tasks across different domains.

[8] Wang, B., & Zhang, Y. (2020). Text-to-SQL Generation for Question Answering on Electronic Medical Records. The study focuses on generating SQL queries from natural language questions specifically in the context of electronic medical records.

[9] Hwang, W., Yim, J., & Lee, H. (2019). A Comprehensive Survey on Text-to-SQL Benchmark Datasets. This survey provides an overview of various benchmark datasets used in the evaluation of text-to-SQL systems.

[10] Shi, T., & Lin, C. Y. (2019). Unsupervised Semantic Parsing with Neural Autoencoders. The authors propose an unsupervised approach to semantic parsing using neural autoencoders, which can be applied to text-to-SQL tasks.

