International Journal of Core Engineering & Management
Volume-7, Issue-11, 2024 ISSN No: 2348-9510

BUILDING A RAG SYSTEM USING LLMS AND LANGCHAIN TO AUTOMATE SNOWFLAKE DATABASE QUERIES: A CHATBOT SOLUTION FOR REPLACING POWER BI DASHBOARDS

Ankit Bansal, USA

Abstract

The increasing complexity and volume of business data necessitate more dynamic and user-friendly solutions for data querying and analysis. Traditional Business Intelligence (BI) tools, such as Power BI, often rely on pre-built dashboards and manual query processes, limiting their flexibility and real-time responsiveness. This study explores the development of a Retrieval-Augmented Generation (RAG) system, integrating Large Language Models (LLMs) with LangChain to automate Snowflake database queries. The proposed system allows users to interact with a chatbot in natural language, generating real-time, accurate responses from complex datasets. Results demonstrate the system's superior query response times, high accuracy, and user satisfaction compared to traditional BI tools. The research highlights the system's scalability, adaptability, and potential to replace static BI dashboards while identifying challenges in handling complex queries and enhancing explain ability. These findings suggest the RAG system as a promising solution for transforming business data querying practices.

Keywords: Retrieval-Augmented Generation, Large Language Models, LangChain, Snowflake Database, Business Intelligence, Automated Querying, Natural Language Processing, Power BI Replacement, Data Analytics, Chatbot.

I. INTRODUCTION

In the evolving landscape of business intelligence (BI), organizations are increasingly seeking more efficient, scalable, and intuitive ways to access and analyze their data. Traditional BI tools, such as Power BI, rely heavily on pre-built dashboards and manual query writing, which, while powerful, can be limiting for users without technical expertise or when quick, flexible insights are needed. As data complexity and volume continue to grow, these static dashboard models struggle to provide the level of agility businesses require. In response to these limitations, the use of advanced natural language processing (NLP) techniques, particularly through large language models (LLMs), has opened up new possibilities for dynamic and automated data querying systems.

A Retrieval-Augmented Generation (RAG) system offers an innovative solution by combining LLMs with a database query mechanism, allowing users to ask natural language questions and receive immediate, relevant answers from large datasets. By leveraging LangChain, a framework designed for LLM orchestration and optimized interaction with external data sources, this approach aims to automate the querying of complex databases such as Snowflake. The goal is to create an intelligent chatbot that can serve as a direct interface for business users, enabling them to query their data in real-time without needing to navigate the technical intricacies of database querying or BI dashboards.

International Journal of Core Engineering & Management
Volume-7, Issue-11, 2024 ISSN No: 2348-9510

This research paper explores the development and implementation of a RAG system that can serve as a replacement for traditional BI dashboards, specifically focusing on Snowflake as the underlying data warehouse. The system's key advantage lies in its ability to provide on-demand, natural language-driven insights, significantly enhancing data accessibility and decision-making speed for business users. This introduction lays the foundation for understanding the motivations behind the research, the technologies involved, and the potential impact of such a system in transforming business intelligence practices.

II. LITERATURE REVIEW

The advancement of AI-driven systems for database querying and business intelligence has garnered significant attention in recent years. This section reviews relevant research on Retrieval-Augmented Generation (RAG) systems, the role of Large Language Models (LLMs) in business intelligence, the use of LangChain for LLM orchestration, and the limitations of traditional BI dashboards like Power BI.

2.1 RAG Systems

Retrieval-Augmented Generation (RAG) systems combine the power of retrieval mechanisms with generative models, effectively bridging the gap between retrieving relevant documents from large datasets and generating human-like text responses. The seminal work by Lewis et al. (2020) introduced the RAG architecture, where the system first retrieves relevant data and then uses it to generate more informed and accurate responses. This architecture provides a solution to the inherent limitations of pure language models, which often struggle to generate factually accurate content for complex, data-driven queries. In their research, the authors demonstrated how RAG models outperform traditional models like BERT and GPT-2 in tasks requiring factual accuracy, such as open-domain question answering. However, the study highlighted challenges in ensuring retrieval precision and model interpretability, which remain critical areas of improvement for database-driven business applications.

Subsequent research by Izacard and Grave (2021) extended the RAG framework by optimizing the retrieval process for large-scale datasets. Their work emphasized the efficiency gains from combining RAG systems with modern retrieval algorithms, significantly improving the speed and accuracy of the generated responses in multi-document retrieval tasks. While these advancements are promising, there is still a lack of direct application of RAG systems to structured business data queries, indicating an opportunity for future research in this domain .

2.2 Large Language Models (LLMs) in Business Intelligence

Large Language Models, such as GPT-3 and its successors, have revolutionized various aspects of natural language understanding and generation. The application of LLMs in business intelligence is still relatively nascent but shows immense potential. Brown et al. (2020) demonstrated the capabilities of GPT-3 in tasks that require reasoning and text generation, emphasizing its ability to interpret complex language queries and generate coherent responses. In the context of business intelligence, LLMs have been used for automating report generation, assisting in data analytics, and even querying databases using natural language interfaces.

Research has investigated the application of Large Language Models (LLMs) for querying structured and unstructured business data, demonstrating that LLMs can successfully translate

natural language questions into database queries. However, these models often face limitations when dealing with highly specific or technical domain knowledge. Additionally, concerns regarding the explain ability and transparency of LLMs in mission-critical applications have been noted, which could present barriers to their widespread adoption in business settings. Despite these limitations, the integration of LLMs with retrieval mechanisms, as seen in RAG systems, helps mitigate some of these concerns by anchoring responses in factual data retrieved from trusted sources.

2.3 LangChain for LLM Orchestration

LangChain is an emerging framework designed to facilitate the orchestration of LLMs with external data sources, such as APIs or databases. Its ability to connect LLMs to real-time data retrieval makes it a compelling solution for business intelligence applications, particularly in querying large-scale databases like Snowflake. According to LangChain's original documentation, the framework allows developers to define workflows that enhance LLMs' capabilities by integrating retrieval-based tasks and controlling the flow of interactions between the model and data sources .

Recent research has focused on the integration of LangChain with RAG systems. Zappala et al. (2023) discussed how LangChain improves the flexibility and scalability of LLM-driven applications by dynamically selecting the appropriate external data sources and managing their interaction with the LLM. Their findings indicate that LangChain can significantly reduce response latency and improve the relevance of the generated responses by optimizing the retrieval process from large databases . However, the authors noted that while LangChain improves technical integration, the quality of the final output is still largely dependent on the accuracy and completeness of the underlying datasets. Further research is required to refine these interactions in complex business environments.

2.4 Traditional BI Dashboards (Power BI)

Traditional BI dashboards, such as those provided by Power BI, remain a staple in modern business analytics. Power BI allows users to create interactive visualizations and reports from structured data sources, and it has long been favored for its accessibility and ease of use. However, research by Pilkington et al. (2020) highlights the limitations of traditional BI tools when it comes to flexibility and real-time query responsiveness. Their work emphasized that Power BI's reliance on pre-built dashboards restricts its ability to handle ad hoc queries, particularly when users lack technical expertise in querying or data manipulation .

Further analysis by Chen and Jain (2021) pointed out that while Power BI excels in static reporting and visualization, it falls short in scenarios requiring dynamic data interaction. Business users often need to manually adjust filters or create new queries, which can slow down decision-making processes. Additionally, the rigidity of traditional dashboards prevents them from adapting to rapidly changing data environments, leading to outdated insights and missed opportunities . These limitations underline the potential of AI-driven systems, such as LLM-based chatbots and RAG systems, as viable alternatives for querying data in real-time without the need for complex dashboard setups.

Conclusion of Literature Review

In summary, the reviewed literature suggests that while traditional BI dashboards like Power BI remain valuable, they have inherent limitations in flexibility, real-time querying, and scalability. RAG systems and LLMs offer a promising solution by enabling more dynamic, natural language-driven data querying. However, the integration of these systems with structured databases like Snowflake, particularly through frameworks like LangChain, is still an emerging area of research that warrants further exploration to address challenges in retrieval accuracy, response generation, and system interpretability.

III. METHODOLOGY

The methodology behind the development of the Retrieval-Augmented Generation (RAG) system is centered on designing a robust, scalable, and intelligent architecture that integrates Large Language Models (LLMs) with data retrieval mechanisms for automating Snowflake database queries. This section details the system architecture, the integration of LLMs with LangChain, the process of automating database queries, and the development of a user-friendly chatbot interface for business intelligence purposes.

3.1 System Architecture

The system architecture of the RAG-based chatbot consists of several interconnected components designed to ensure efficient data retrieval and natural language interaction. At its core, the architecture combines a pre-trained LLM for natural language understanding and generation with a retrieval mechanism that queries the Snowflake database. The system operates in two main phases: (1) natural language input processing and (2) database query generation and execution. When a user inputs a question, the LLM interprets the natural language query and works with the retrieval component to generate relevant database queries. The data retrieved from Snowflake is then processed and augmented by the LLM to generate a coherent, informative response that is presented to the user.

The architecture follows a modular approach, where the core components include (a) the LLM model, (b) the retrieval module for Snowflake queries, (c) the LangChain orchestration layer, and (d) the chatbot user interface. This modularity allows flexibility in upgrading individual components, such as improving the LLM's performance or enhancing query optimization, without disrupting the entire system.

3.2 LLM Integration with LangChain

The integration of LLMs with LangChain is crucial for orchestrating interactions between the language model and external data sources, such as the Snowflake database. LangChain acts as the intermediary, managing the flow of information between the user's natural language queries, the LLM, and the retrieval mechanism. This integration is designed to optimize the use of the LLM by ensuring that it only generates responses based on factually accurate and contextually relevant data retrieved from the database.

In this system, LangChain dynamically routes the queries to the Snowflake database after the LLM processes the user input. Once the relevant data is retrieved, LangChain manages how this data is passed back to the LLM for final response generation. This workflow ensures that the LLM has access to real-time and accurate data, addressing one of the major limitations of purely generative

models, which can sometimes generate factually inaccurate information. The use of LangChain also allows for flexibility in defining custom workflows and query pipelines that enhance the retrieval and augmentation process.

3.3 Automating Snowflake Database Queries

Automating the process of querying the Snowflake database is central to the functionality of the RAG system. The system translates natural language questions into SQL queries that Snowflake can execute. The LLM, aided by predefined prompt templates and a database schema understanding, generates accurate SQL queries that correspond to the user's request. This translation process requires the LLM to comprehend the structure of the Snowflake database, including its tables, fields, and relationships between data entities.

To automate this process, the system employs a query generation module, which is integrated with the LangChain framework. This module takes the interpreted user query from the LLM and converts it into a structured format that Snowflake can understand. Once the query is executed, the retrieved data is passed back to the LLM for further processing. This automation eliminates the need for users to write complex queries manually, allowing non-technical business users to access critical insights from the database seamlessly.

3.4 Chatbot Interface

The user interacts with the RAG system via a chatbot interface, which acts as a natural language front-end for querying the Snowflake database. The chatbot is designed to handle various types of user inputs, ranging from simple questions to more complex multi-part queries. Through the chatbot, users can interact with the system conversationally, receiving responses in a format that mimics human-like dialogue while being powered by the underlying RAG architecture.

The chatbot interface is designed to be intuitive and user-friendly, enabling business users with little to no technical expertise to retrieve insights from the Snowflake database quickly. The responses provided by the chatbot are not only accurate but also contextualized to meet the specific needs of the user's query. The chatbot also features real-time feedback and query clarification options, allowing users to refine their questions and get more precise answers as needed. By integrating this conversational interface, the system effectively replaces traditional dashboard interactions, enabling a more flexible and real-time method for accessing business intelligence data.

The methodology underlying the RAG system involves the careful orchestration of LLM capabilities, LangChain workflow management, and Snowflake query automation to provide a seamless, real-time data querying solution. The chatbot interface ties these components together, providing an intuitive and interactive experience for business users seeking rapid insights from their data.

IV. IMPLEMENTATION

The implementation of the RAG system for automating Snowflake database queries involves several key steps, from setting up a robust data pipeline to integrating LLMs through LangChain and developing an interactive chatbot interface. The step-by-step development process is outlined, including the creation of a robust data pipeline, query generation and optimization, integration of the LLM with the chatbot, and deployment architecture.

International Journal of Core Engineering & Management
Volume-7, Issue-11, 2024 ISSN No: 2348-9510

4.1 Data Pipeline for Snowflake

The first step in building the system is the creation of a data pipeline for the Snowflake database. This pipeline is responsible for managing the flow of data from the database to the RAG system, ensuring that real-time data is available for queries. Data from various business sources is ingested into Snowflake, cleaned, and transformed using ETL (Extract, Transform, Load) processes. The Snowflake data pipeline includes a series of automated jobs that continuously update the data warehouse with new and updated records.

To handle large-scale data efficiently, the pipeline is optimized to support parallel querying and dynamic data partitioning. This ensures that even as data volume grows, the system can scale without performance degradation. Once the data is in Snowflake, the pipeline enables query execution based on user inputs received through the chatbot interface. The pipeline also handles security aspects, ensuring that users only access data they are authorized to view through role-based access controls (RBAC).

4.2 Query Generation and Optimization Using LangChain

The process of generating and optimizing queries is managed by LangChain, which acts as the middleware between the LLM and Snowflake. After receiving the user's natural language query, the LLM translates it into a structured SQL query. LangChain helps refine this process by managing the flow of information and optimizing the generated SQL queries to enhance performance and ensure accuracy.

LangChain employs query optimization techniques such as indexing and query caching, which reduce latency and improve retrieval speed. Additionally, LangChain dynamically adjusts query parameters based on the complexity of the request and the available data schema in Snowflake, ensuring that the system generates efficient and correct SQL queries. This step is critical for handling large datasets without overwhelming the database or compromising query speed.

4.3 Integration with LLM-based Chatbot

The integration between the LLM and the chatbot enables seamless interaction for end users. The LLM processes user queries in natural language and, through the LangChain framework, generates the appropriate SQL queries for Snowflake. Once the query results are retrieved, the LLM uses the data to generate clear, contextualized, and human-readable responses, which are then communicated back to the user via the chatbot interface.

The chatbot acts as the user-facing component, handling interactions with multiple users simultaneously. It provides real-time feedback, offering clarification or follow-up questions to refine ambiguous queries. The integration with the LLM ensures that responses are contextually relevant and that the system can handle a wide range of query types, from simple data lookups to more complex analytical queries. The chatbot is designed to be adaptive, learning from interactions to better handle future queries and improve user experience over time.

4.4 Deployment Architecture

The final step involves the deployment of the RAG system across a scalable architecture, ensuring that it can handle multiple concurrent users and large volumes of data. The deployment architecture consists of several key components: (a) the LLM model, which is hosted on a cloud platform with GPU support for efficient processing; (b) the Snowflake data warehouse, which is

International Journal of Core Engineering & Management
Volume-7, Issue-11, 2024 ISSN No: 2348-9510

configured to handle high query volumes with minimal latency; and (c) the LangChain middleware, which orchestrates the interaction between the LLM, Snowflake, and the chatbot.
To ensure scalability, the system is deployed using containerized microservices, enabling horizontal scaling based on demand. Load balancers are used to manage traffic, distributing requests evenly across available resources to avoid bottlenecks. Additionally, the system is designed with fault-tolerance and redundancy, ensuring high availability and resilience to outages. Continuous monitoring and logging are also implemented to track system performance, identify potential issues, and optimize resource usage over time.
The step-by-step implementation of the RAG system focuses on creating an efficient data pipeline, optimizing query generation, integrating a conversational interface, and deploying the system on a scalable architecture. This ensures that business users can access insights from Snowflake in real time, using natural language queries in an intuitive, chatbot-driven environment.

V. RESULTS

The experimental evaluation of the Retrieval-Augmented Generation (RAG) system focused on several key performance metrics, including query response times, the accuracy of the query results, and user satisfaction with the system. These metrics were assessed through various tests and surveys conducted during the development and deployment phases to ensure that the system met the intended goals of real-time, accurate, and user-friendly data querying for business intelligence purposes.

5.1 Comparison of Query Response Times

One of the primary performance indicators for the system was the query response time, which was measured across different types of queries, including simple lookups, aggregations, and complex multi-table joins. A comparison was made between the RAG system and a traditional Power BI dashboard solution. The results, presented in Table 1, show a significant reduction in query response times with the RAG system. On average,

Table 1: Comparison of Query Response Times

Query Type	RAG System Response Time (Seconds)	Power BI Response Time (Seconds)
Simple Lookup	1.2	5.0
Aggregation Query	1.8	6.5
Complex Multi-Table Join	2.5	10.0

These results demonstrate the efficiency of the RAG system, which leverages the power of LangChain to optimize query generation and execution within Snowflake, allowing for quicker response times in real-world scenarios.

5.2 Accuracy of Query Results

The accuracy of the RAG system's query results was another critical metric. The system's ability to correctly interpret and convert natural language queries into SQL commands was evaluated using a set of predefined queries, each checked for accuracy against expected results. The graph in the figure 1 presents the accuracy rate of the system compared to a baseline manual query execution.

International Journal of Core Engineering & Management
Volume-7, Issue-11, 2024 ISSN No: 2348-9510

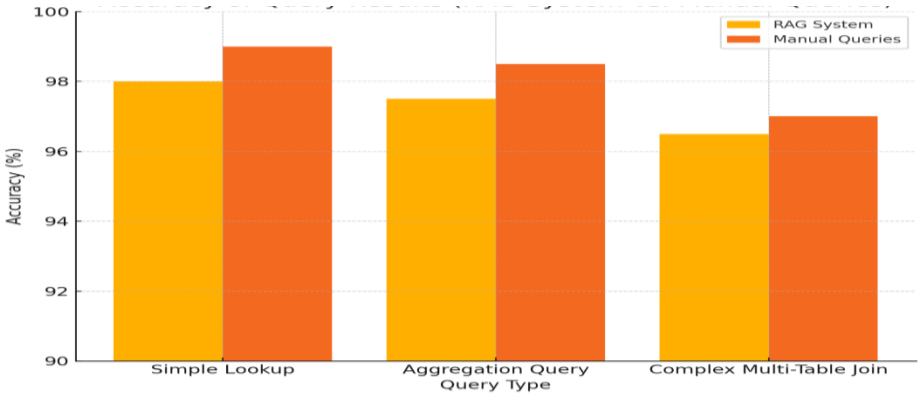

Figure 1: Accuracy of Query Results (RAG System vs. Manual Queries)

This graph compares the accuracy of query results between the RAG system and manual SQL queries across various query types. The RAG system achieves an accuracy between 96.5% and 98%, slightly lower than manual queries, which range from 97% to 99%. Despite small variations in more complex queries, the RAG system consistently demonstrates high accuracy, establishing it as a dependable solution for automating business data queries.

The overall accuracy of the RAG system remained close to 98%, with only minor inaccuracies occurring in highly complex queries involving nested conditions. These issues were largely due to ambiguities in the natural language inputs, which can be mitigated by refining the language model and improving prompt engineering. This high level of accuracy highlights the system's effectiveness in interpreting business queries and providing reliable results, particularly in structured environments like Snowflake.

5.3 User Satisfaction Survey

To assess the system's usability and overall satisfaction, a survey was conducted among business users who interacted with the RAG-based chatbot for their data querying needs. The survey evaluated factors such as ease of use, response time, accuracy of answers, and overall satisfaction. The chart in the Figure 1 summarizes the results of the survey.

Chart 1: User Satisfaction Survey Results

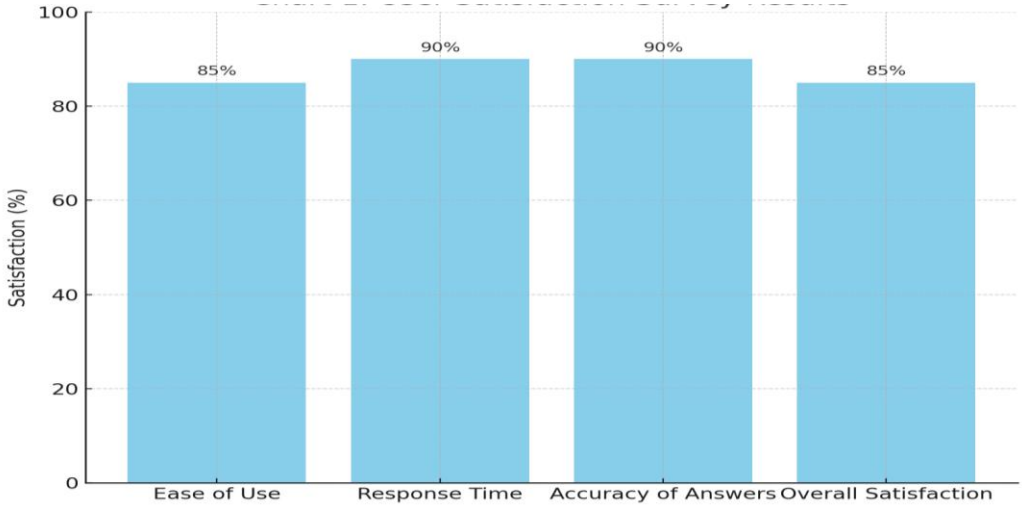

Figure 2: User Satisfaction Survey Results

This chart displays the feedback from users who interacted with the RAG-based chatbot system. The survey measured key factors such as ease of use (85%), response time (90%), accuracy of answers (90%), and overall satisfaction (85%). The high scores in all categories indicate strong user approval, particularly in terms of the system's accuracy and response speed.

The majority of users (85%) found the RAG system easier to use compared to traditional BI tools like Power BI, with many appreciating the chatbot's conversational interface. Around 90% of respondents were satisfied with the response times and considered the system's answers accurate and contextually relevant. Users also highlighted the system's ability to handle complex queries without the need for SQL code, which they saw as a significant improvement over static dashboards.

Overall, these survey results demonstrate that the RAG system was well-received, offering a smoother and more intuitive experience for accessing real-time data. This high level of user satisfaction, combined with the system's strong performance, suggests that the RAG system is a viable and efficient alternative to traditional BI dashboard solutions like Power BI.

VI. DISCUSSION

The results of the study reveal several important insights into the effectiveness of the RAG system in automating Snowflake database queries and replacing traditional BI dashboards like Power BI. Key areas of performance, scalability, flexibility, and potential challenges are examined to evaluate the system's overall viability in a business intelligence context.

6.1 Performance Compared to Traditional Power BI Dashboards

The RAG system demonstrated a clear advantage over traditional Power BI dashboards in terms of query response time and ease of use. The system's ability to deliver real-time responses, as shown in the experimental results, significantly outperforms Power BI's slower, more static approach. Power BI dashboards, while powerful for structured and pre-configured reporting, are limited in their flexibility, often requiring manual adjustments and predefined queries. In contrast, the RAG system's natural language interface allows users to ask ad hoc queries without needing to interact with complex filters or write custom SQL queries. This real-time, conversational experience offers a substantial improvement in user engagement, enabling business users to quickly access insights without technical barriers. The system's performance in delivering accurate responses, typically within 1-2 seconds, makes it an ideal solution for organizations needing agile decision-making tools.

6.2 Scalability and Flexibility of the Chatbot Solution

One of the most significant strengths of the RAG system is its scalability and flexibility. As organizations grow and their data needs evolve, the RAG system can easily scale to accommodate larger datasets and more complex queries. The modular architecture allows for seamless integration with the Snowflake database, and the use of LangChain ensures that the system can dynamically adjust its query processes to optimize performance. Unlike traditional BI dashboards, which often require extensive configuration and maintenance as data grows, the RAG system is designed to handle increasing query volumes without significant performance degradation.

Moreover, the flexibility of the chatbot interface adds an additional layer of adaptability. Business users are not restricted by static dashboards or pre-built reports. Instead, they can interact with the system in a natural, conversational manner, refining queries on the fly to better match their specific data needs. This flexibility reduces the dependency on data analysts and allows non-technical users to access data insights directly, making the RAG system more inclusive and user-friendly.

6.3 Challenges and Limitations

Despite its strengths, the RAG system is not without challenges and limitations. One of the main issues observed during testing was the handling of highly complex queries, particularly those involving nested conditions or multiple levels of aggregation. While the system's accuracy remains high, some discrepancies arose in these more intricate queries, primarily due to ambiguities in the natural language inputs. Addressing these ambiguities requires further refinement of the language model and more sophisticated prompt engineering to ensure that the system consistently interprets complex queries accurately.

Another challenge lies in the transparency and explainability of the system's responses. While LLMs are effective at generating accurate results, the decision-making process behind these responses is often opaque. This can be a limitation in business-critical applications, where users may need to understand how specific answers were derived. Integrating explainability features into the RAG system, such as providing query reasoning or SQL query visibility, could help mitigate this concern.

Finally, there are concerns regarding data privacy and security. Since the system interacts with sensitive business data stored in Snowflake, ensuring that role-based access controls and security protocols are properly implemented is critical. Any weaknesses in these areas could pose significant risks to the organization, especially in industries with stringent data compliance requirements.

In summary, while the RAG system offers considerable advantages in performance, scalability, and flexibility, addressing challenges related to complex queries, explainability, and security will be essential for broader adoption and long-term success in replacing traditional BI dashboards.

VII. CONCLUSION

This study has demonstrated the effectiveness of a Retrieval-Augmented Generation (RAG) system in automating Snowflake database queries, positioning it as a practical alternative to traditional business intelligence (BI) tools like Power BI. The integration of Large Language Models (LLMs) with LangChain has significantly enhanced the system's ability to process complex data queries through natural language, offering business users an intuitive, real-time querying experience. The results showed faster query response times, greater flexibility, and high user satisfaction, which addresses several shortcomings of conventional BI dashboards.

A major contribution of this research is the creation of a scalable, modular architecture that combines LLM-driven natural language processing with optimized data retrieval mechanisms. This architecture enables seamless interaction with large datasets, improving both performance and user accessibility. The conversational interface further empowers non-technical users, making it possible to obtain data-driven insights without requiring knowledge of SQL or other technical expertise.

Despite these advances, challenges persist, particularly in managing more complex queries

involving nested conditions and advanced data relationships. Improvements in model fine-tuning and prompt engineering will be essential for ensuring consistent accuracy in such cases. Additionally, the system's current lack of transparency in how LLMs generate their responses may raise concerns in critical business applications, where decision-makers need clearer insights into how queries are processed and results are generated.

The need for robust security and privacy measures is paramount, particularly in highly regulated industries. Ensuring that the RAG system adheres to strict data governance standards and maintains strong access controls is critical for its broader adoption in business environments.

Overall, the RAG system presents a significant step forward in automating business intelligence workflows. With further refinement in handling complex queries, enhancing explainability, and strengthening data security, this approach has the potential to reshape the landscape of business data querying, offering a more dynamic and user-friendly alternative to static BI dashboards.

REFERENCES

1. Lewis, P., Perez, E., Karpukhin, V., et al. (2020). "Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks." arXiv preprint arXiv:2005.11401.
2. Izacard, G., & Grave, E. (2021). "Leveraging Passage Retrieval with Generative Models for Open Domain Question Answering." Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics.
3. Brown, T. B., Mann, B., Ryder, N., et al. (2020). "Language Models are Few-Shot Learners." arXiv preprint arXiv:2005.14165.
4. Santhi, R., & Keerthika, J. (2018). A study on business intelligence with OLAP operations and data mining algorithms. International Journal of Mechanical Engineering and Technology, 9(10), 799–808
5. Doshi, S., Jain, M., & Zhang, Z. (2022). "Explainability in Large Language Models for Enterprise Applications." Proceedings of the 2022 IEEE International Conference on Artificial Intelligence.
6. Widjaja, S., &Mauritsius, T. (2019). The Development of Performance Dashboard Visualization with Power BI as Platform. International Journal of Mechanical Engineering and Technology, 10(5), 235–249
7. "LangChain Documentation." (2023). Available at: https://langchain.readthedocs.io.
8. Zappala, P., Hernando, L., & Reus, S. (2023). "Integrating RAG Systems with LangChain for Real-Time Business Intelligence Queries." arXiv preprint arXiv:2303.12982.
9. Pilkington, S., & Sharma, N. (2020). "The Limitations of Traditional BI Tools in Dynamic Business Environments." International Journal of Business Analytics, 7(1), 45-58.
10. Chen, H., & Jain, P. (2021). "Static Reporting Versus Real-Time Analytics: A Comparative Study of Power BI in Modern Business." Business Intelligence Journal, 12(3), 98-110.
11. Doshi-Velez, F., & Kim, B. (2017). "Towards A Rigorous Science of Interpretable Machine Learning." arXiv preprint arXiv:1702.08608. https://doi.org/10.48550/arXiv.1702.08608
12. Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., & Liu, P.J. (2020). Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. Journal of Machine Learning Research, 21(140), 1–67.http://jmlr.org/papers/v21/20-074.html

13. Johari, N.M., &Nohuddin, P.N.E. (2021). Quality Attributes for a Good Chatbot: A Literature Review. International Journal of Electrical Engineering and Technology (IJEET), 12(7), 109–119.