GoNoGo V2: Leveraging LLMs for Automated Data Analysis in Automotive Software Release Decisions: An Industrial Case Study

***Index Terms*—software testing, industrial scenario, tabular data analysis, large language model**

I. Introduction

In this project, we aim to develop an intelligent data analysis platform GoNoGo using large language models (LLMs) to streamline the testing and validation process in automotive software development. In the automotive industry, releasing a new software, such as for autonomous driving, requires seamless integration with various hardware and software components, including critical systems like braking and acceleration. Before deployment, all supporting functionalities must be thoroughly verified to ensure safety and reliability. Currently, testing engineers manually update dynamic testing results into databases and perform time-consuming analyses to assess system readiness. Our platform automates this process by responding to natural language queries, retrieving and analyzing relevant data from the database, and providing actionable insights, thus significantly enhancing efficiency and decision-making.

GoNoGo faces several challenges as follows.

1. One of the primary challenges is the heterogeneity between natural language queries and structured tabular data in the database. The system needs to accurately interpret user queries, which may refer to attributes in the database using slightly different terminology. For example, a query might ask for information on a column labeled ”name,” while the database contains this information under ”names.” Bridging this semantic gap requires advanced natural language processing (NLP) techniques to map query terms to corresponding table columns, ensuring accurate retrieval of relevant data and enhancing system adaptability.

2. Another challenge is ensuring the system can recognize and filter out irrelevant or non-actionable queries. To prevent the system from generating misleading or incorrect responses, it must identify queries that fall outside its operational scope and provide users with feedback on what kinds of questions are valid. This capability is essential for maintaining the system’s reliability and guiding users toward effective interactions, minimizing the potential for confusion or errors during the decision-making process.

3. Our database is highly dynamic, with constant updates and new content being added regularly. As a result, the LLM-based system must not only be able to adapt to the evolving data landscape but also generalize to new domain-specific information. This dynamic nature introduces a layer of complexity in maintaining the system’s accuracy and relevance as the underlying data grows and shifts.

4. The real-world automotive context demands an exceptionally high degree of precision. Errors, such as hallucinations or inaccurate results, could have serious implications in critical decision-making processes. In the automotive industry, especially in safety-critical applications like autonomous driving or vehicle control systems, the margin for error is virtually nonexistent. Therefore, ensuring that the system provides reliable and accurate outputs is of utmost importance, requiring robust mechanisms to detect and prevent erroneous results.

GoNoGo introduces several contributions.

1. We have developed a novel sample generation method that automatically creates a diverse set of few-shot examples. These examples significantly enhance the performance of the system by providing richer training data compared to the original natural language queries.

2. GoNoGo has been successfully deployed within a leading global automotive manufacturer, where it is currently in active use by approximately 200 people. This large-scale deployment demonstrates the robustness and scalability of our platform in a real-world industrial setting, showcasing its potential to streamline the software release process by automating complex data analysis tasks.

II. Related work

[1]

III. Conclusions

References

[1] OpenAI, “Introducing structured outputs in the api.” https://openai.com/index/introducing-structured-outputs-in-the-api/, 2023. Accessed: 2024-09-23.