SQL Query Generation with Transformer API

Objective

The goal of this lab was to evaluate the performance of the defog/sqlcoder-7b-2 Transformer model in generating SQL queries from natural language prompts. The evaluation involved crafting three distinct questions, observing the generated SQL queries, and analyzing the results for correctness and relevance based on a given database schema.

Prompt Versions and Results

Version 1

Question: "Show the revenue per product for sales made in New York during the previous month."

Result: The model successfully generated a correct SQL query using appropriate joins between the sales, products, and salespeople tables. Revenue was computed accurately using the expression price \* quantity. The query included proper date filtering and grouping.

Version 2

Question: "List each product and the total revenue it generated last month in the New York region, sorted by revenue in descending order."

Result: The generated SQL included accurate logic for computing total revenue and ordering results in descending order. The use of grouping and aggregation was correct, and the query reflected the intent of the natural language question.

Version 3 (Edge Case)

Question: "What is the average customer satisfaction score by product sold in the New York region last month?"

Result: The model returned the response SELECT 'I do not know' AS answer;. This was the expected and correct outcome because the database schema does not include a customer satisfaction field. The model correctly identified the lack of schema support for the query and avoided generating a hallucinated or invalid SQL statement.

Observations

The model consistently produced correct and executable SQL queries for realistic business questions.

The handling of the edge case in Version 3 demonstrates the model's schema awareness and reliability.

The formatting of the output using sqlparse improved readability and presentation.

The quality and specificity of the prompt significantly influenced the clarity of the generated query.

Conclusion

The sqlcoder-7b-2 model demonstrated strong capabilities in translating natural language to SQL within the constraints of a defined schema. It was able to handle varying question phrasings, generate accurate joins and aggregations, and appropriately decline to answer unsupported queries. These qualities make it a promising tool for natural language interfaces in data analysis and business intelligence applications.