Visuo Home Assignment Challenge

The objective of this challenge was to build a text to SQL query system that accepts a natural language question and responds with a valid SQL query that should help in answering the question in context of a database with respect to which the questions are to be asked.

There are two main modules developed as part of the solution:

1. Experiment: This is a module to test and validate various prompts, LLMs, LLM configurations and visualize the results in an experiment tracking tool – weights and biases (wandb.ai).
2. Inference: A FastAPI supported backend to serve the users request and prompt the LLM which is hosted locally or hosted in the cloud or any other 3rd party service.

Dataset used:

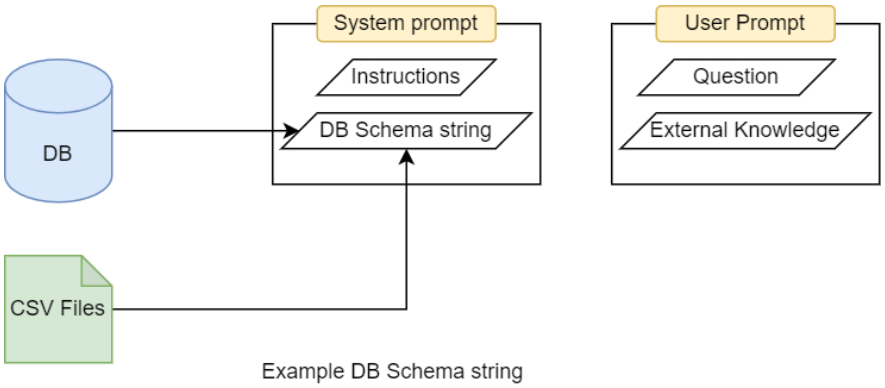
The annotated dataset I have used for this challenge is the BIRD-SQL mini dev dataset (https://bird-bench.github.io/) which is a very popular open source benchmarking dataset consisting of 1154 high-quality examples of **questions** and **SQL** pairs along with the **sqlite database** on which the SQL can be verified and an **external knowledge** that can help in writing the SQL query.

There are 3 difficulty levels present in this dataset: simple, moderate and challenging, I have evaluated the model on all difficulty. In the next section we will see how the prompt is designed using all this information.

This website also contains a leaderboard showcasing the top language model that has been benchmarked on this dataset.

Prompt Construction:

The system prompt contains the instructions that instructs the LLM to generate the SQL query for the user's request and it also contains the DB Schema string which is built using the csv files



Example DB Schema string

```
CREATE TABLE Books (
BookID INT PRIMARY KEY Unique identifier for each book
Title VARCHAR(255) NOT NULL Title of the book,
Author VARCHAR(100) NOT NULL Name of the author);
```

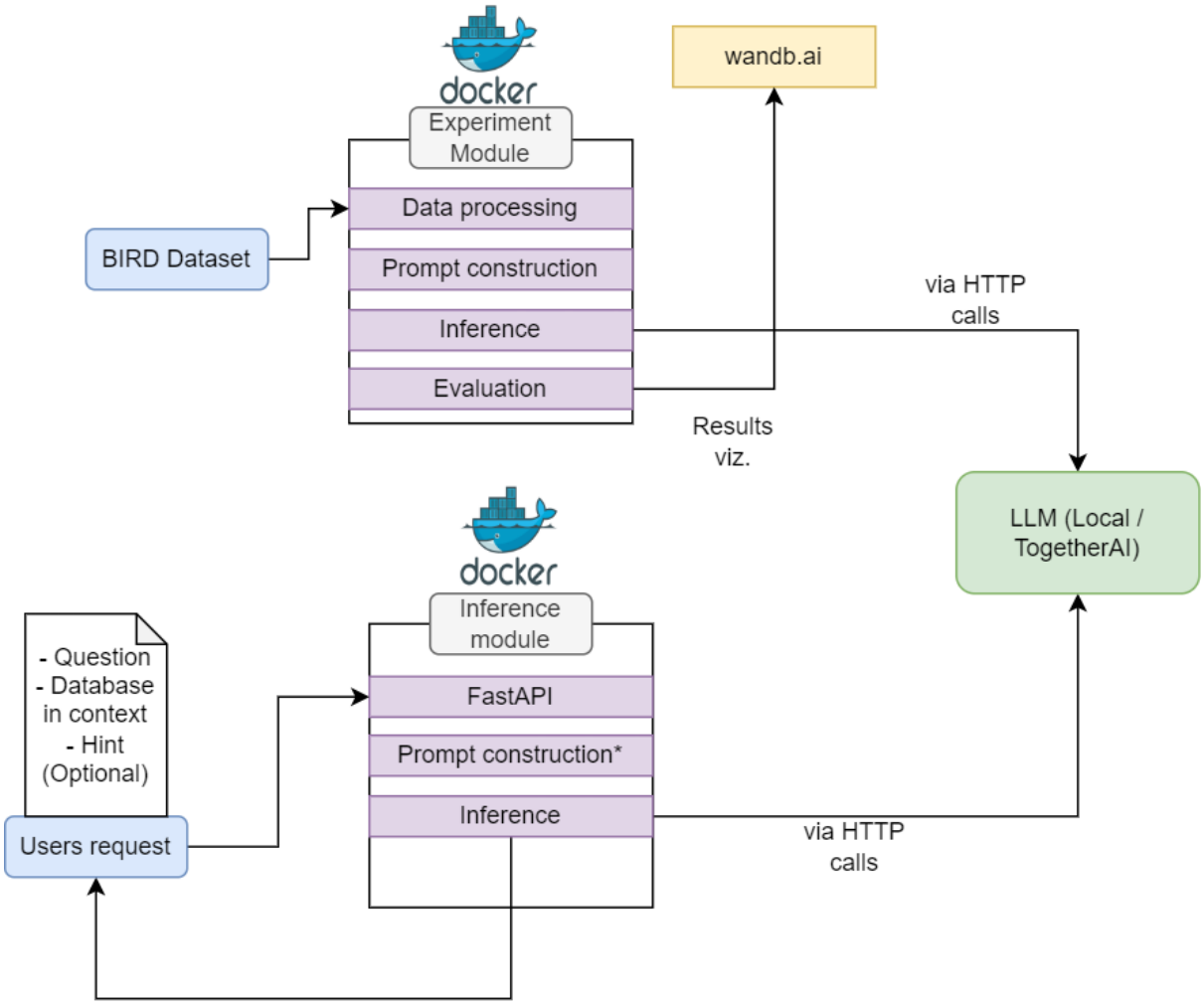
containing the column description and all the tables in the database in context.

The user prompt contains the question along with a hint or external knowledge that helps in answering the question.

Solution design

Below is the architecture of the solution provided, below are some explanations for the key design decisions:

1. Experiment module: Contains the code for testing and validating the prompt using the method explained here . The data processing part builds the key elements required to construct the prompt i.e. create the database schema string that includes the column description as well. The data models are defined in data.py using pydantic to ensure data integrity.
2. Inference module: Contains the FastAPI backend code to take the user's request convert it to a prompt using the same logic defined in experiment module and respond with LLM's response.
3. Each module is containerized using docker to port the code to any cloud environment.



Evaluation:

The metrics used to measure the performance on this dataset is the standard metric used of benchmarking on this dataset i.e. execution accuracy. To achieve this the SQL query generated from the LLM response is executed on the SQLite database and if the result is equal to that of ground truth query, then it is considered as the true positive and the accuracy is computed by taking the ratio of true positive over total number of data points.

I have done some quantitative & qualitative analysis which can be reproduced with the github code.

Quantitative analysis:

Model Name	Difficulty Level	Execution Accuracy	# of Data points
meta-llama/Meta-Llama-3.1-70B-Instruct-Turbo	simple	0.6352	925
meta-llama/Meta-Llama-3.1-70B-Instruct-Turbo	moderate	0.4859	464
meta-llama/Meta-Llama-3.1-70B-Instruct-Turbo	challenging	0.4326	145
meta-llama/Meta-Llama-3-70B-Instruct-Turbo	moderate	0.2343	464
meta-llama/Meta-Llama-3-70B-Instruct-Turbo	challenging	0.2340	145

Note: Above results are obtained with the best performing prompt design.

Qualitative analysis:

Examples of correct predictions (simple):

Question	Type	Query	result
What is the highest eligible free rate for K-12 students in the schools in Alameda County?	AI Generated SQL Query	SELECT MAX(CAST(REPLACE(`Free Meal Count (K-12)`, ',', '') AS REAL) / CAST(REPLACE(`Enrollment (K-12)`, ',', '') AS REAL)) FROM frpm WHERE `County Name` = 'Alameda'	Correct
What is the highest eligible free rate for K-12 students in the schools in Alameda County?	Gold Standard SQL Query	SELECT `Free Meal Count (K-12)` / `Enrollment (K-12)` FROM frpm WHERE `County Name` = 'Alameda' ORDER BY (CAST(`Free Meal Count (K-12)` AS REAL) / `Enrollment (K-12)`) DESC LIMIT 1	
Please list the phone numbers of the schools with the top 3 SAT excellence rate.	AI Generated SQL Query	SELECT T1.Phone FROM schools AS T1 INNER JOIN satscores AS T2 ON T1.CDSCode = T2.cds ORDER BY CAST(T2.NumGE1500 AS REAL) / CAST(T2.NumTstTakr AS REAL) DESC LIMIT 3	Correct
Please list the phone numbers of the schools with the top 3 SAT excellence rate.	Gold Standard SQL Query	SELECT T1.Phone FROM schools AS T1 INNER JOIN satscores AS T2 ON T1.CDSCode = T2.cds ORDER BY CAST(T2.NumGE1500 AS REAL) / T2.NumTstTakr DESC LIMIT 3	

Examples of Incorrect predictions (simple):

Since the foreign key information is missing in the implementation that is why the below query is not giving the expected results:

Question	Type	Query	result
For the female client who was born in 1976/1/29, which district did she opened her account?	AI Generated SQL Query	SELECT T3.A2 FROM client AS T1 INNER JOIN disp AS T2 ON T1.client_id = T2.client_id INNER JOIN district AS T3 ON T3.district_id = T1.district_id WHERE T1.gender = 'F' AND T1.birth_date = '1976/1/29'	Incorrect
For the female client who was born in 1976/1/29, which district did she opened her account?	Gold Standard SQL Query	SELECT T1.A2 FROM district AS T1 INNER JOIN client AS T2 ON T1.district_id = T2.district_id WHERE T2.birth_date = '1976-01-29' AND T2.gender = 'F'	

Proposal for further improvements

1. **Context improvement using conversational memory**: By utilizing the conversational memory features provided by the Langchain framework, the system can retain and recall previous interactions, thereby improving the accuracy and relevance of the generated SQL queries. This approach allows the model to build upon prior context, making it particularly useful for handling complex queries that require multi-turn conversations or when the initial SQL query fails to execute correctly.
2. **Version control for validated prompts:** To ensure consistent and reliable performance of the text-to-SQL system, it is crucial to implement version control for the prompts used during the generation process. This can be achieved by maintaining a repository of validated prompts, which can be systematically updated and versioned. These artifacts can then be uploaded to Weights and Biases (wandb.ai. In the production environment, the inference module can download the appropriate versions of the prompts, ensuring that the system always uses the most effective and tested configurations. This practice will help in maintaining the integrity and reliability of the system over time.
3. **Feedback loop:** Creating a robust feedback loop is essential for continuously monitoring and improving the system's performance. This involves regularly measuring execution accuracy and tracking the performance metrics against predefined benchmarks.
4. **VectorDB implementation:** Developing a vector database to store historical user interactions can significantly enhance the system's ability to provide relevant recommendations and insights. By indexing and storing previous queries and their corresponding SQL outputs, the system can leverage this historical data to identify patterns and recommend similar analyses to users.
5. **Schema-Aware Parsing:** Integrate more sophisticated methods for parsing and utilizing database schema information to enhance the accuracy of generated SQL queries, especially in handling joins and complex queries involving multiple tables.