Report on the Work Done

March 26, 2025

GitHub Link

`https://github.com/DanieleFerneti/bird_schema_alignment`

1 Problem Analysis

The work focused on analyzing and evaluating the alignment between natural language questions and SQL database tables used to answer these questions. The objective was to verify the ability of an LLM to correctly identify the relevant tables for an SQL query based on natural language queries and compare its predictions with the actual values extracted directly from SQL queries.

2 Methodologies

2.1 Dataset Used: Bird-benchmark(dev)

The chosen dataset, Bird-benchmark(Mini-dev), is a reduced benchmark, provided by the link given in the assignment but efficient for evaluating and developing solutions related to this task. It contains 500 pairs of SQL queries and natural language questions, allowing us to test the model's accuracy.

The first step was to create a Python script, `ask_tables.py`, which allowed obtaining the reference tables for each SQL query in the dataset both through regular expressions and using a large language model.

2.1.1 Extraction of Actual Tables

A function was implemented that uses regular expressions to analyze SQL queries and identify the names of the tables used.

- For each of the 500 queries, the list of tables was extracted and saved in a JSON file.

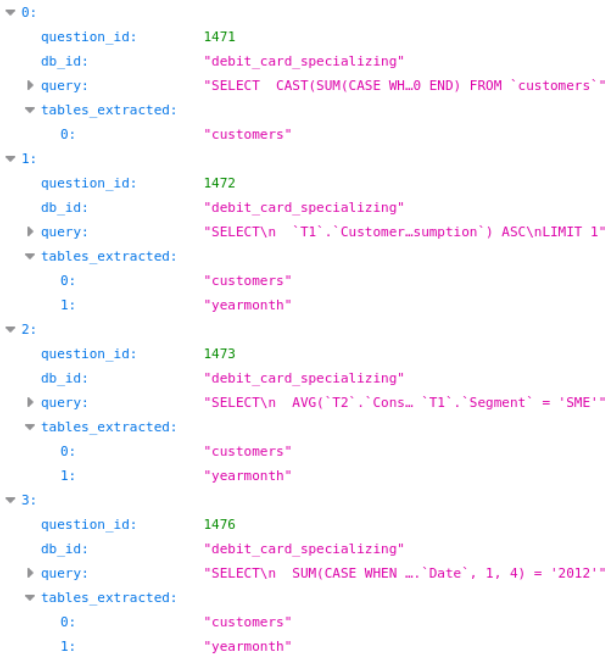

Figure 1: Example of tables extracted from SQL queries using regular expressions

2.1.2 Prediction of Tables with LLM

Another function was created to query the Llama 3-70B model (API Groq) asking it to identify the relevant tables based on the natural language question.

- To improve response accuracy, the model was provided with context based on the concept of schema alignment:
  - The tables extracted from the corresponding SQL query were provided.
  - Additionally, the names of tables from other databases where these tables were present were also included.
- These predictions were also saved in a JSON file.

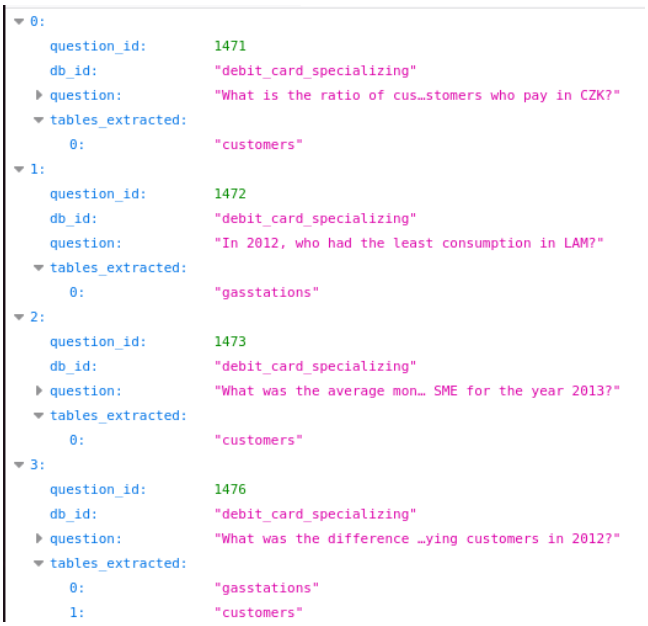

Figure 2: Example of tables extracted from SQL queries using LLM

To avoid issues with the API Groq request limit (30 requests per minute), queries were sent in batches of 30, with a 60-second wait between batches.

2.2 Evaluation Metrics Calculation

After obtaining the two JSON files (actual and predicted tables), a Python script was created to evaluate the LLM's performance.

2.2.1 Calculation of Precision, Recall, and F1 Score

To compare the actual tables with the predicted ones, the following indicators were calculated:

- **True Positive (TP)** → Correctly predicted tables
- **False Positive (FP)** → Incorrectly predicted tables by the model
- **False Negative (FN)** → Actual tables not identified by the model

The metrics were calculated using standard formulas:

$$\text{Precision} = \frac{TP}{TP + FP}$$

$$\text{Recall} = \frac{TP}{TP + FN}$$

$$\text{F1-score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

For each query, both global values and those for each database (db_id) were calculated.

```
debit_card_specializing:    0.7605633802816902
student_club:               0.7288135593220338
thrombosis_prediction:      0.5957446808510638
european_football_2:        0.6333333333333333
formula_1:                  0.794392523364486
superhero:                  0.7702702702702702
codebase_community:         0.8439306358381503
card_games:                 0.8794326241134751
toxicology:                 0.5688073394495413
california_schools:         0.8767123287671232
financial:                  0.7850467289719627
```

Figure 3: F1 for each single database

```
precision:    0.7086183310533516
recall:       0.8131868131868132
f1_score:     0.7573099415204679
```

Figure 4: Global Precision, Recall, and F1

2.3 Comparison with Metrics Calculated by LLM

In addition to manual calculation with Python, the Llama 3-70B model was asked to calculate Precision, Recall, and F1-score by providing it with the TP, FP, and FN values as input.

The results obtained from the LLM were almost identical to those manually calculated, with only minor approximations.

Metric	**Calculated**	**LLM**
Precision	0.7086	0.7080
Recall	0.8132	0.8130
F1-score	0.7573	0.7570

Table 1: Comparison between manually calculated global metrics and those calculated by the Llama 3-70B model

3 Results

3.1 Visualization of Results

Finally, a histogram was created to graphically represent the F1-scores of each database (db_id).

This allowed visualizing performance trends for each database and identifying possible trends or critical points.

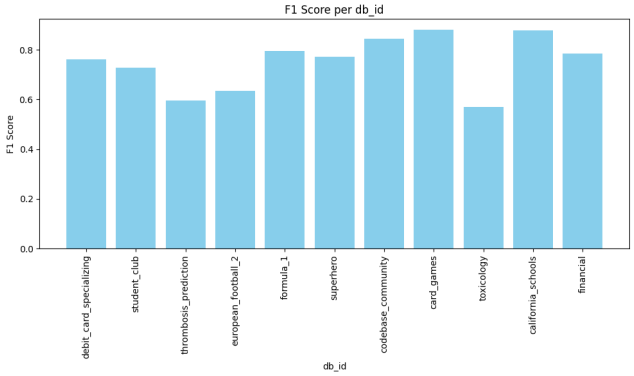

Figure 5: F1 score histogram

3.2 Final Considerations

- The LLM was able to accurately identify the correct tables based on the natural language question.
- The comparison between actual and predicted values showed that the model has a good semantic alignment with the database.
- Using the API Groq enabled leveraging a powerful model (Llama 3-70B) while respecting rate limits.
- The quantitative analysis with evaluation metrics provided a clear measure of system accuracy.
- The bar chart made database-specific performance more readable, highlighting possible improvements.

Conclusion: The adopted method, combining regex extraction, LLM prediction, and validation with standard metrics, effectively tested the model's ability to perform schema alignment in the SQL context.