

2nd Round Task for Thomas Haid

Business analytics is concerned with generating data-driven summaries for decision-making. The data required for these reports is stored in relational databases and SQL is used to retrieve the data. With the advent of LLMs there is now the possibility to formulate a reporting task in natural language and let the required SQL statements being generated by an AI. This makes the data easily accessible and should allow the business analyst to focus on the data questions instead of thinking about SQL statements.

The goal of this task is to develop a natural language (NL) interface for generating SQL queries to ease the search in [The History of Baseball](#) database. For this, you first need to create a prompting concept and then define a suitable way to test the results to identify regression or improvements whenever there are changes applied to the initial prompt.

Please be prepared for a total presentation time of 1 hour.

[Part A] Development of a prompt (theoretical part)

Create a **prompting concept to produce SQL queries from NL** for execution on the database. Consider how a prompt for such queries could look like including the instruction, the context describing the database, user input, etc.

As an example, you can consider the test set of sample questions (e.g. based on KaggleDBQA <https://github.com/chiahsuan156/KaggleDBQA>).

Prepare a written prompt template with concrete text and placeholder blocks, as well as one specific example of text how it would be sent to the LLM. Explain which sections of the prompt contain what information, why it is relevant, and what prompting concepts were used.

[Note] This part is considered theoretical to avoid the necessity of having an actual LLM endpoint at hand, which is usually associated with costs, however, try to be specific wrt. the wording and structures that are being used and explain how a potential implementation e.g. with templates could look like.

[Part B] Implementation of a test approach for generated queries (practical part)

Imagine that the prompt in Part A was implemented and it is now possible to create SQL statements from NL. The file `examples_queries_test.json` contains a set of exemplary NL queries and the expected SQL response (ground truth) for testing, while `generated_queries.json` contains the LLM generated pairs. Unfortunately, the generated response only partially matches the expected SQL queries.

The goal of this part is to **investigate the validity of the differences** and **develop a test framework** (e.g. using python) that allows for the **assessment of the differences in an automated way** and **can be reused for various evaluation runs**.

Imagine the resulting comparison code to be deployed in a CI/CD pipeline to monitor the generated queries after a code change to identify if the change was an improvement or not.

Show and explain the test framework you developed, present examples, and provide reasons for your implementation choices.

Notes and guidance:

- Being specific is better than talking at high level
- The preparation time is limited, no need to be perfect
- If helpful, slides, mockups or sketches are welcome
- The workshop is interactive, not a one-way presentation