CS 224V Assignment 2

Due: 10/16, 2:00 PM PST (Part 1 due on Wednesday 10/11, 11:59pm PST)

**Instruction:** Use this Colab notebook in conjunction with this write-up. Make sure to “Save a copy in Drive” before running the notebook. Submit your answers through Gradescope and attach your Google Colab notebook. **In red, we label how each question in this writeup corresponds to a Gradescope question.**

This assignment is designed to be completed in **groups of 2.** Please submit as a group to Gradescope.

**Extensions:** You are granted an automatic 24-hour extension to either Assignment 1 or Assignment 2, but not both. As explained earlier, if any of your group members have used this extension on Assignment 1, then your group does not have this extension on Assignment 2.

1 **Project Proposals**

If you are looking to work on a mentor-written proposal, please fill out Mentor-Written Project Interest Form by Wednesday 10/11, 11:59pm PST. This is an earlier deadline to allow mentors to have ample time to assign you to proposals.

If you are looking to do a custom proposal, please fill out Custom Project Intent Form by Monday 10/16, 2:00pm PST

You are welcome to both apply to mentor-written proposals as well as to brainstorm your own custom proposal, but you must at least do one of these options. (Gradescope Q1)

2 **Introduction**

BIRD [1] is a dataset to evaluate LLM’s performance on text-to-SQL tasks, and as the authors reported, the performance is sub-optimal for all of the LLMs on the market. In this homework, you will explore how to prompt LLMs to generate SQLs and gain hands-on experience on evaluating its performance on this dataset. Bird-Bench is composed of over 12.5K unique question-SQL pairs at a diversity of complexities and in several different domains. The goal is to be able to convert natural language into schema-aware and complex SQL queries, effectively enabling one to traverse large databases and ground agents in concrete knowledge.

2.1 **LLM model**

Thanks to Google’s generous support for our course, every students gets a Google credit of $50 for the course. You can use PaLM-2 LLM in this assignment. Please follow the instructions in the Colab on redeeming the course credit and setting up API access.

However, it was reported that OpenAI’s models deliver better results than PaLM-2 on this benchmark. The recently released gpt-3.5-turbo-instruct is much more affordable than GPT-4. If you prefer to use OpenAI models for this homework, albeit at your own expense, you are welcome. Please clearly state which LLM you are using, so we can collect the results for different models, compare the results, and share them with the class. Your grade will not be affected by which model you use; we will evaluate the homework based on how you improve the chosen model with prompts.
What is the LLM you are using?  (Gradescope Q2.1)

2.2 Domain selection

Similar to HWK 1, sign up on this sheet on which domain you’d like to experiment with. Domains here are from the dev set of BIRD. As in homework 1, the choice of your domain has no impact on your grades.

What is the domain of your choice?  (Gradescope Q2.2)
3 Zero-shot prompting

To begin with, let’s see the model’s capability to generate SQL with only schema descriptions. In this section, you will construct one prompt for all questions in your domain.

3.1 Prompt construction

A zero-shot prompt should contain the following parts:

1. A system message (e.g., “You are a semantic parser. Generate a query for a restaurant database with the following signature.”)

2. The schema description. The common way to include this is by supplying it with the `CREATE TABLE` command that was used to create this table. If there are multiple tables in the database, we can concatenate the descriptions together.

At run-time for each question, the user’s query will also be included in the prompt.

An example prompt can be found on slide 29 of the Lecture 3 slides.

3.2 Hand evaluation of your engineered prompt

Follow the instructions on the Colab to evaluate your prompt on 6 examples.

(a) What are the prompt and the model’s answer for those 6 examples? (Gradescope Q3.1 - Q3.6)

(b) Comment in 2-3 sentences in your writeup on how well the model accomplishes the task of zero-shot SQL query generation with these hand-evaluated prompts. Does the model struggle to generate the valid SQL string more so as the question becomes more challenging? (Gradescope Q4)

3.3 Batch evaluation on your chosen domain

Follow the instructions on the Colab to evaluate your prompt on all examples in your domain.

(a) What is the accuracy score you got from this domain? (Gradescope Q5)
4 Few-shot prompting

One common method to improve the performance of LLMs is to provide them with few-shot examples. Provide two examples from the domain of your choice, and evaluate the model’s performance. In this section, you will construct one prompt for all questions in your domain.

4.1 Prompt construction

A few-shot prompt should contain the following parts:

1. A system message (e.g., “You are a semantic parser. Generate a query for a restaurant database with the following signature.”)

2. The schema description. The common way to include this is by supplying it with the `CREATE TABLE` command that was used to create this table. If there are multiple tables in the database, we can concatenate the descriptions together.

3. A few examples to go along with the schema description. In this homework, you will choose two examples from the dataset and include those in the prompt. (Note: you should use the same few-shot examples for all questions in your domain.)

At run-time for each question, the user’s query will also be included in the prompt.

4.2 Hand evaluation of your engineered prompt

Follow the instructions on the Colab to evaluate your prompt on 6 examples.

(a) What are the prompt and the model’s answer for those 6 examples? (Gradescope Q6.1 - Q6.6)

(b) Comment in 2-3 sentences in your writeup on how well the model accomplishes the task of few-shot SQL query generation with these hand-evaluated prompts. Does the model struggle to generate the valid SQL string more so as the question becomes more challenging? (Gradescope Q7)

4.3 Batch evaluation on your chosen domain

Follow the instructions on the Colab to evaluate your prompt on all examples in your domain.

(a) What is the accuracy score you got from this domain? (Gradescope Q8)
5 Going beyond few-shot

Even with few-shot examples, the model still struggles to predict correct SQLs for most of the dataset points. In this section, try out one or multiple of the following methods:

- More complicated few-shot examples or more few-shot examples (at most 5 examples);
- Implement Chain-of-Thought prompting [2] (Note: this requires a separate prompt to extract the result);
- Include field descriptions;
- Include table descriptions;
- Other methods that you can think of.

Iterate through several variations of the prompt and report on the following:

(a) **The best batch evaluation score that you achieved.** (Gradescope Q9)

(b) **What steps you took to improve the prompts beyond our basic few-shot setting, to achieve this score.** Comment in 1-2 sentences on why you believe your improvement helped the model better perform on this task. (Gradescope Q10)

(c) **Perform error analysis on the first 6 errors.** (Gradescope Q11.1 - 11.6)

For each failure case, look up the gold query from the dataset and write 1 sentence on why the predicted SQL is incorrect (e.g. “the predicted column name Enrollment (Ages 5-17) is wrong”). Then, write 1 sentence on why do you think the model failed (e.g. “the wording for user’s query creates confusion on the correct column name”).

6 **Remember to upload your Colab notebook as a PDF** (Gradescope Q12)
References

[1] Jinyang Li, Binyuan Hui, Ge Qu, Binhua Li, Jiaxi Yang, Bowen Li, Bailin Wang, Bowen Qin, Rongyu Cao, Ruiying Geng, Nan Huo, Chenhao Ma, Kevin C. C. Chang, Fei Huang, Reynold Cheng, and Yongbin Li. Can llm already serve as a database interface? a big bench for large-scale database grounded text-to-sqls, 2023.