Generate SQL queries from Natural Language: Project Improvements

The project was done in 5 hours including writing this improvement report and organising readme for better understanding. Note that the results or the work shown in this report is not optimal. In order to finish the task I used only the more feasible options. More optimized results can be produced with more research and exploration. Here's how I would improve the challenge outcome if I had more time.

1. FINE-TUNE LLM

1.1 DATASET

Use multiple datasets. Some examples below:

- https://huggingface.co/datasets/gretelai/synthetic_text_to_sql (**most** 6.5k downloads, 106K rows)
- https://huggingface.co/datasets/Clinton/Text-to-sql-v1 (800 downloads, 262k rows)
- https://huggingface.co/datasets/hardikch05/100000_text_to_sql (**most** rows: 78M)

Create synthetic data or scrape data from the web.

1.2 FRAMEWORK

I took inspiration from [1][2] since I already followed it before. However, it is outdated now. But following the official documentation is relatively easy.

Although there are many possibilities to accomplish the task. I would prefer a library that provides the maximum functionality. Below are some examples:

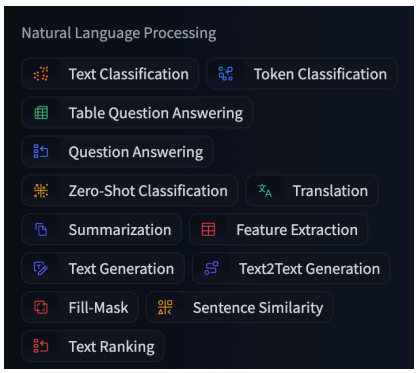
- HuggingFace (https://huggingface.co/docs/trl/en/index) is used for this task. I chose HF over Unsloth since I am already experienced in HF and wanted to finish the project within 3-4 hours.
- Unsloth (https://github.com/unslothai/unsloth): GPU version not supported in Mac m2. Possible solution is to use Google colab for fine-tuning.

In the case of innovation or customized use of functions, I would prefer to use PyTorch and write the custom functions with better improvements. That will give me more flexibility. But this custom approach is more suitable for academia and not really suitable for production.

1.3 MODEL

- Limitations: On Local machine (max m2) I cannot use the 4 bit models since bitsandbytes isn't natively supported on mac non-cuda devices.
- How to choose the best performing model: OpenLLM Leaderboard: https://huggingface.co/open-llm-leaderboard

- I’ve decided to use a text-to-text since this fits the task the most.



- There exist already fine-tuned model for text2sql: suriya7/t5-base-text-to-sql. Based on t5-base model. In future this model can used to compare the performance of the model that I fine tuned.
- Note: There already exist fintuned text2sql models which will perform better than my model since mine is trained on less data and for less time.

1.4 FINE-TUNE

- Start with a small dataset [3]
- Once the fine-tune worked on the small dataset, I fine-tuned the model on a relatively larger dataset.
- However, using the full dataset and additionally other data sources will improve the model performance further.

1.5 TRAINING RESULTS

- Find the loss and mean token accuracy below in Fig 1.

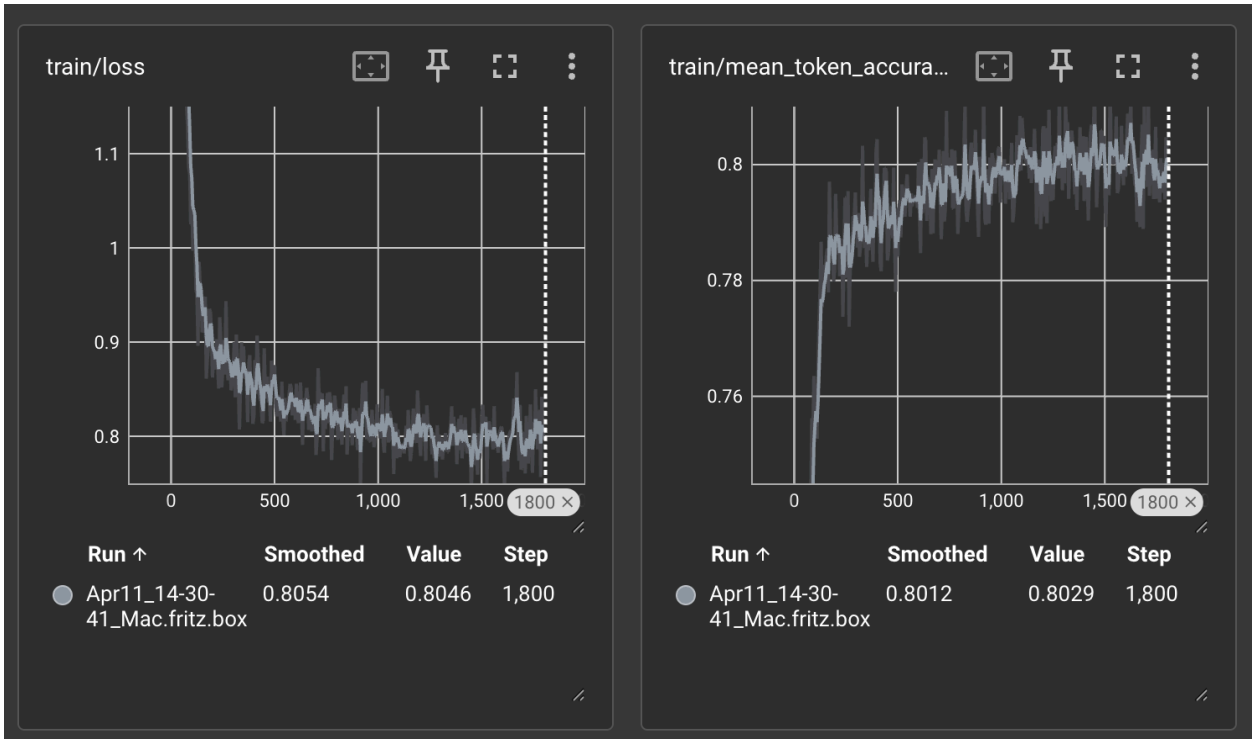


Fig 1: Tensorboard loss and accuracy logs.

2. DEPLOY FINE-TUNED MODEL

- The model is deployed on the HuggingFace hub.
- It's deployed as a gradio [5] chatbot app.
- You can use the deployed model here: https://huggingface.co/spaces/rat45/sql-sft-lora-model

3. TARGET MACHINE

- Initially tried with Google Colab. However, constant disconnection and failure to save models delayed my work. So I had to switch to my local machine.
- My target machine was a Mac M2 with 16 GB memory.
- Using a good GPU instance with approx. 24GB will increase the training speed significantly. Unsloth will also improve the training speed on supported machines.

4. TESTING DEVICE

The app [4] is tested on the following platforms:

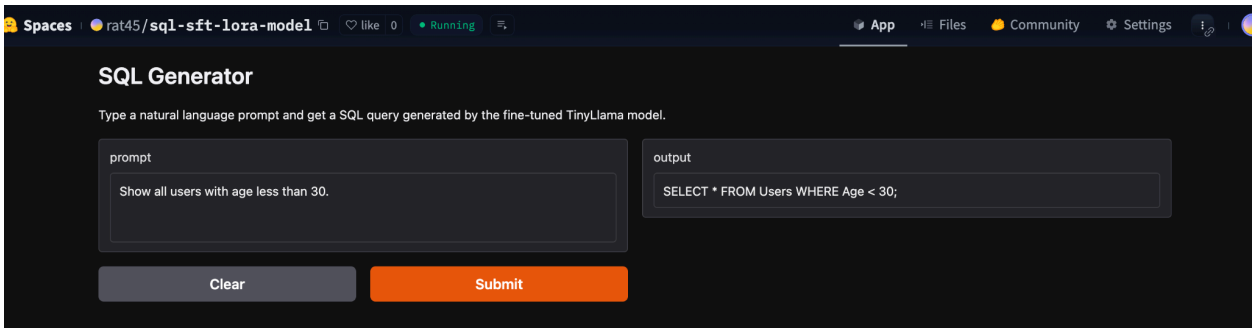
- PC: Google chrome browser (Ran with no issues)
- PC: Vivaldi browser (Didn't work)
- Phone: IPhone default web browser: (Be careful that if the phone screen goes black that would interrupt the process)
- Starting the app take some warmup time
- Given the prompt, generating the query takes from 1 to 2 minutes since the app is running on CPU.

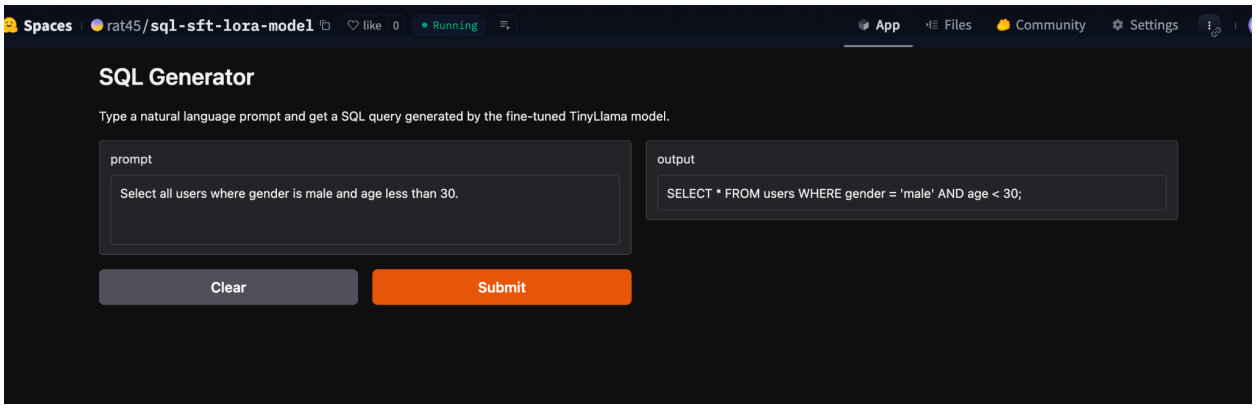
REFERENCES

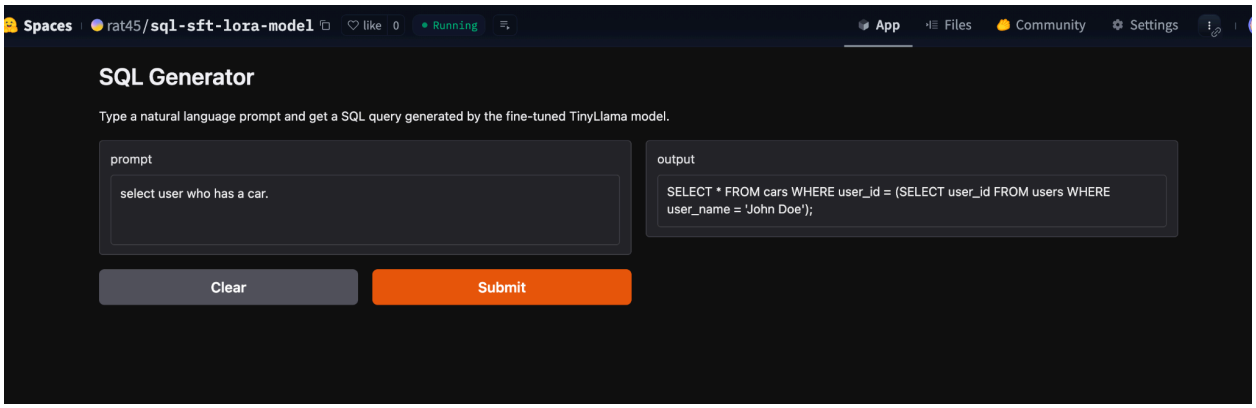
[1] https://github.com/huggingface/alignment-handbook/blob/main/scripts/run_sft.py
[2] https://huggingface.co/docs/trl/en/sft_trainer
[3] https://karpathy.github.io/2019/04/25/recipe/
[4] https://huggingface.co/spaces/rat45/sql-sft-lora-model
[5] https://www.gradio.app/

RESULTS

Below are snippets of some of the results.







Results can also be bad sometimes which can be improved using the techniques mentioned earlier in this report: