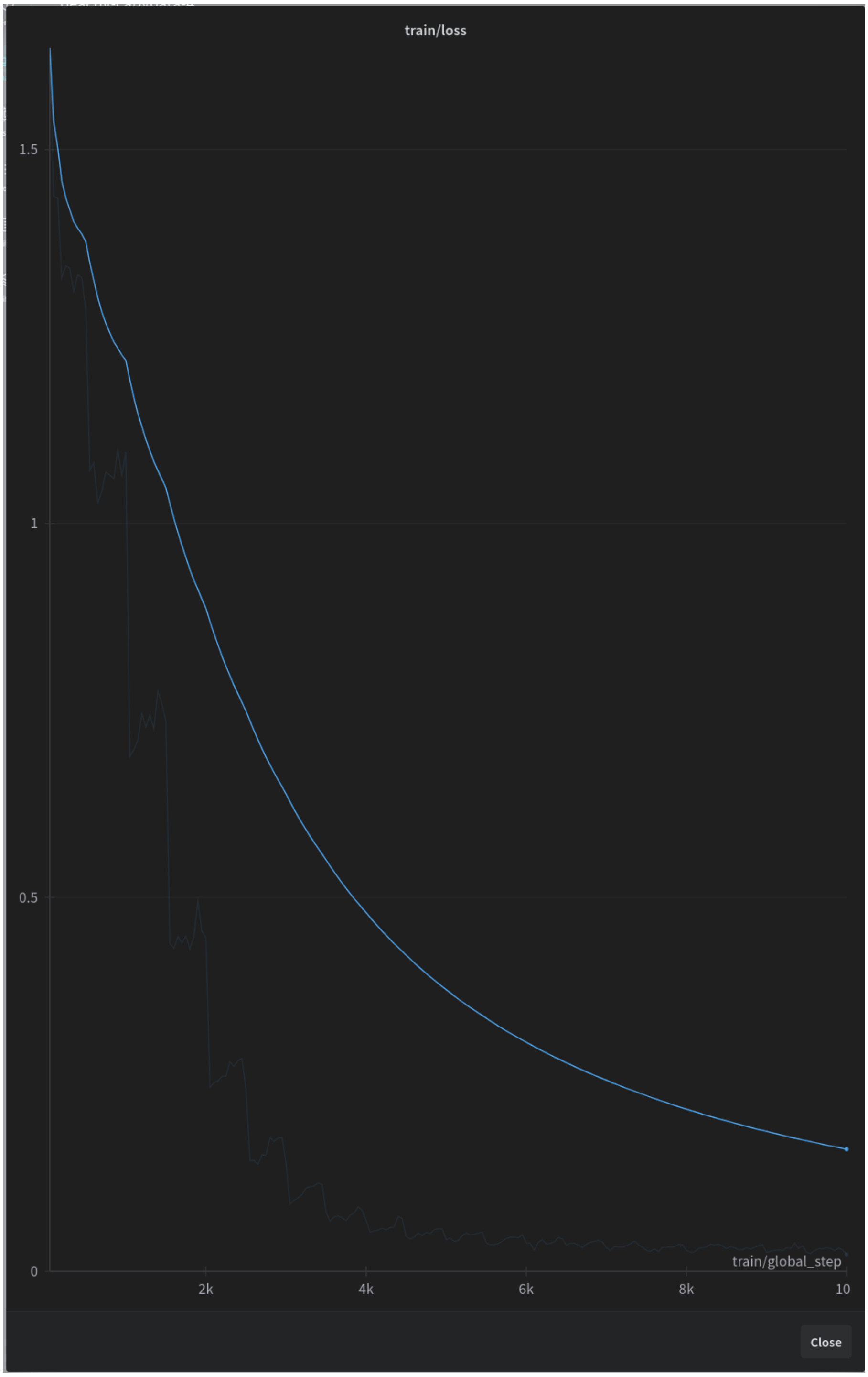
NTU ADL Homework 3, 2023 Fall.

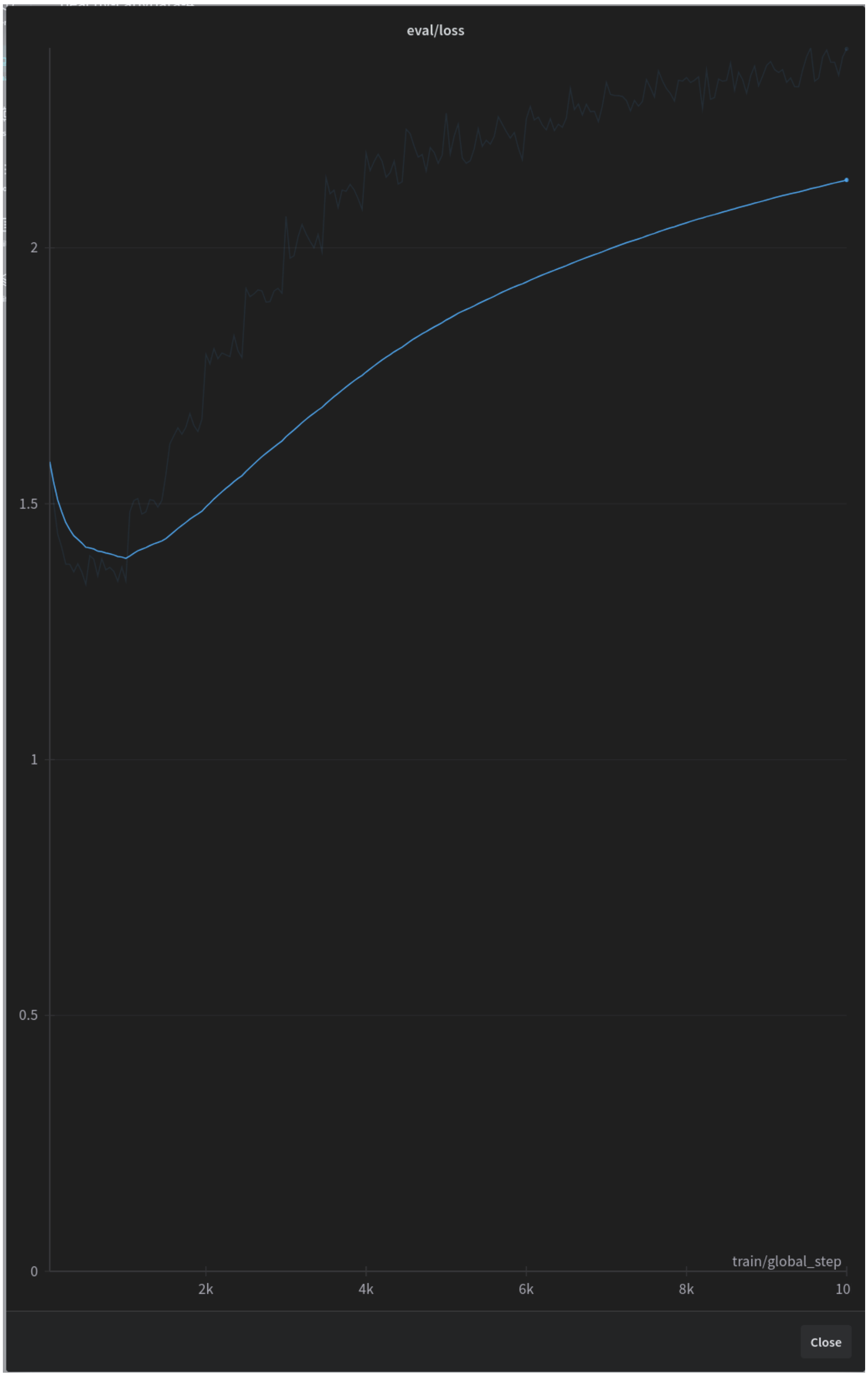
LLM Tuning

Describe & Performance:

- **I used all the training data while training, at first, I didn’t think too much, just normally trained it with all training data, it took 10k training steps, at that time, I decided to evaluate every 500 steps, followed by this thought, something happened…**



- **Above picture is my training loss curve, it looks pretty well, right? Let me show you another figure…**



- **This picture is my evaluation loss curve, it looks like a NIKE logo, however, it's not a good symbol…**

- **Combined these two figures, it shows that my adapter was overfitting after about 1.1k steps, I've found this situation at that time, however, I just decided to finishing this training period, and it took me about 110 hours.**

- **Following picture is my training hyper-parameters.**

```
python train.py \
    --model_name_or_path /home/cjweibb/workspace/2023_NTUADL/testSpace/Taiwan-LLM-7B-v2.0-chat \
    --output_dir ./output/test\
    --logging_steps 50 \
    --save_strategy steps \
    --data_seed 42 \
    --save_steps 500 \
    --save_total_limit 40 \
    --evaluation_strategy steps \
    --eval_dataset_size 1024 \
    --max_eval_samples 1000 \
    --per_device_eval_batch_size 1 \
    --max_new_tokens 32 \
    --dataloader_num_workers 1 \
    --group_by_length \
    --logging_strategy steps \
    --remove_unused_columns False \
    --do_train \
    --do_eval \
    --lora_r 64 \
    --lora_alpha 16 \
    --lora_modules all \
    --double_quant \
    --quant_type nf4 \
    --bits 4 \
    --warmup_ratio 0.03 \
    --lr_scheduler_type constant \
    --gradient_checkpointing \
    --dataset /home/cjweibb/workspace/2023_NTUADL/testSpace/data/train.json \
    --source_max_len 128 \
    --target_max_len 128 \
    --per_device_train_batch_size 1 \
    --gradient_accumulation_steps 16 \
    --learning_rate 0.0003 \
    --adam_beta2 0.999 \
    --max_grad_norm 0.3 \
    --lora_dropout 0.01 \
    --weight_decay 0.0 \
    --seed 42 \
    --dataset_format alpaca \
    --report_to wandb
```

- **Following picture is my model's performances on public_test set:**

```
Loading checkpoint shards: 100%|                                                  [7/12]
                                                   | 2/2 [00:05<00:00,  2.72s/it]
100%|
                                              | 250/250 [03:24<00:00,  1.22it/s]
Qlora-finetuned Inference results:
Mean perplexity: 3.4397967281341555
(ntu_adl) cjweibb@mvclab-cjw:~/workspace/2023_NTUADL/testSpace$ bash eval.sh
Loading checkpoint shards: 100%|
                                                   | 2/2 [00:05<00:00,  2.99s/it]
100%|
                                              | 250/250 [03:29<00:00,  1.19it/s]
Qlora-finetuned Inference results:
Mean perplexity: 3.6434353761672975
(ntu_adl) cjweibb@mvclab-cjw:~/workspace/2023_NTUADL/testSpace$ bash eval.sh
Loading checkpoint shards: 100%|
                                                   | 2/2 [00:05<00:00,  2.73s/it]
100%|
                                              | 250/250 [03:33<00:00,  1.17it/s]
Qlora-finetuned Inference results:
Mean perplexity: 5.025580849647522
(ntu_adl) cjweibb@mvclab-cjw:~/workspace/2023_NTUADL/testSpace$ bash eval.sh
Loading checkpoint shards: 100%|
                                                   | 2/2 [00:05<00:00,  2.67s/it]
100%|
                                              | 250/250 [03:32<00:00,  1.17it/s]
Qlora-finetuned Inference results:
Mean perplexity: 17.62995908117294
```

- **As you can see, there are four perplexity scores listed from top to bottom, corresponding to ordered checkpoints: "500 steps, 1500 steps, 2000 steps, 10000 steps." The results confirm what I mentioned earlier about "overfitting," and my best performance is at 500 steps with a perplexity score of 3.439.**

LLM Inference Strategies

Zero-Shot, Few-Shot:

Zero-Shot inference :

- **For Zero-Shot inference, I just directly inply “Taiwan-LLM-7B-v2.0-chat” model on public_test dataset with max_length=2048, below picture shows my custom prompt, there is a little different with original one given by TAs.**

```python
def get_prompt(instruction: str) -> str:
    '''Format the instruction as a prompt for LLM.'''
    return f"你是人工智慧助理，以下是用戶和人工智能助理之間的對話。你要對用戶的問題提供有用、安全、詳細和禮貌的回答。\n請回答下面的問題。\nUSER: {instruction} ASSISTANT:"
```

- **Following screenshot is my mean perplexity score of Zero-Shot Inferencing:**

```
Mean perplexity: 5.452533513069153
```

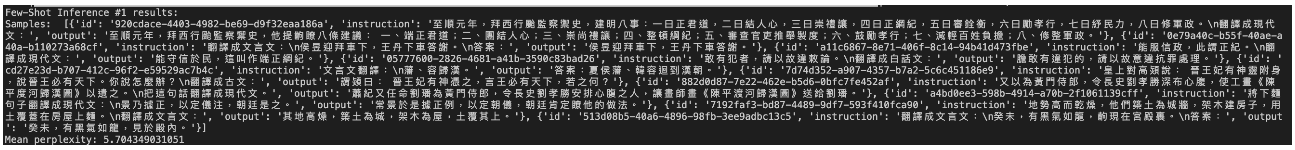
Few-Shot inference :

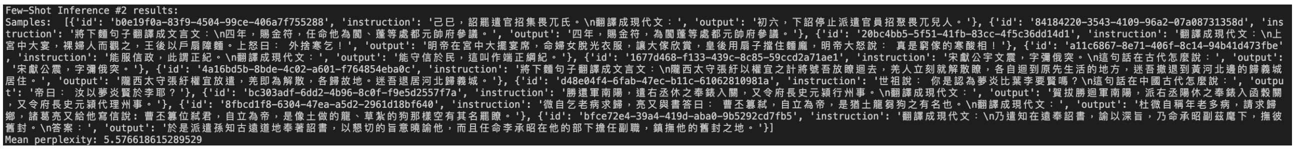
- **For Few-Shot inference, I just directly inference “Taiwan-LLM-7B-v2.0-chat” model on public_test dataset with max_length=2048, and I have done an experiment, following are my experiment steps:**
    - **1. Randomly Select 10 different datas from public_test dataset as Few-Shot prompts’ instructions and outputs.**
    - **2. Remove datas that have been selected as Few-Shot prompts.**
    - **3. Modify the prompt as shown below.**

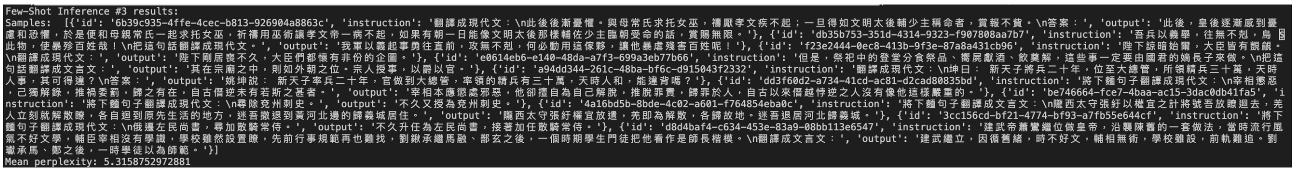
```
你是人工智慧助理，以下是用戶和人工智能助理之間的對話。你要對用戶的問題提供有用、安全、詳細和禮貌的回答。
以下為幾個範例：
用戶：左衛大將軍宇文述勸說煬帝選挑選幾驪名精銳騎兵突圍齣去，納言蘇威說： 據守城池我方還有餘力，而輕騎則是對方的長處，陛下是萬乘之主，怎麼可以輕易行動？
幫我把這句話翻譯成文言文 助理：左衛大將軍宇文述勸帝簡精銳數驪騎潰圍而齣，納言蘇威日： 城守則我有馀力，輕騎乃彼之所長，陛下萬乘之主，豈宜輕動！
用戶：盧延貴非常害怕，急急忙忙逃走，他卻連連呼籲： 不要害怕，我是個人！
這句話在中國古代怎麼說： 助理：見人即行起而來，延貴懼而卻走。此物連呼： 無懼，吾乃人也。
用戶：秦頌是都人，從江夏都尉升任南陽太守，上任時經過宜城城內，看見一傢朝東的房子，他停車觀看，說：這個住人的地方可以築墓。
翻譯成文言文： 助理：頡，都人也，以江夏都尉齣為南陽太守。徑宜城中，見一傢東嚮，頡住車視之，日：此居處可作塚。
用戶：不用兵而巧設騙局取勝，他就是這樣做的。南蠻
翻譯成古文： 助理：而驃信讋慓，不假兵以詐勝，斯之謂也。南蠻
用戶：這是二不可。
這句話在古代怎麼說： 助理：不可二。
用戶：有的說不吃葷腥，不近女色，遇上這樣的好藥就能成為地仙。
把這句話翻譯成文言文： 助理：或不葷血，不色欲，遇之必能降真為地仙矣。
用戶：四人祝壽完畢，趨步離開。
這句話在古代怎麼說： 助理：四人為壽已畢，趨去。
用戶：六月，西瓜、甜瓜、蓮子、饕瓜。
翻譯成白話文： 助理：六月，獻西瓜、甜瓜、蓮子、饕瓜。
用戶：翻譯成現代文：
上遣使者諭釋，冠蓋相望。 助理：唐僖宗接連不斷地派遣使臣前往李剋用處進行規勸解釋。
用戶：孝文貞皇後林氏，是平原人。
幫我把這句話翻譯成文言文 助理：孝文貞皇後林氏，平原人也。
請根據以上的範例進行回答。
用戶：翻譯成文言文：
於是，廢帝讓瀋慶之的堂侄、直將軍瀋攸之賜瀋慶之毒藥，命瀋慶之自殺。 助理：
```

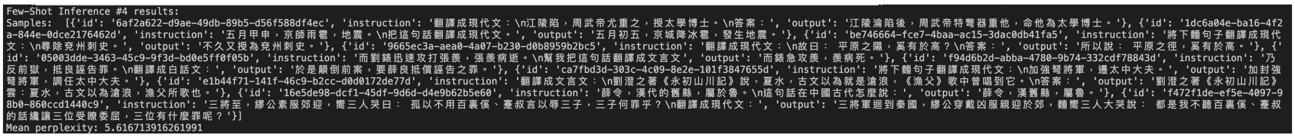
    - **4. Repeat above steps for 5 times.**

- **Following figures are my Few-Shot inference perplexity scores:**









- **Upon observing the results and the samples I selected as prompts, I've noticed that if the samples we choose are too imbalanced, the performance tends to be poorer.**

- **After examining the dataset, it can be broadly categorized into two main branches: "Classical Chinese translated into Modern Chinese" and "Modern Chinese translated into Classical Chinese." The imbalance refers to a situation where the samples are skewed towards a particular direction,this will affecting the model's understanding of the prompt.**

- **On another note, I have also observed that the accuracy and level of detail in the text content impact the output in some aspects. Accuracy here pertains to the correctness of the text itself or the accuracy of the meaning of words.**

Comparison:

- **After observing the results of Zero-Shot and Few-Shot approaches, we can see that Few-Shot does indeed achieve better results than Zero-Shot based on the provided samples.**

- **However, we also notice from the perplexity results that the impact of samples on Few-Shot Inference is significant. Therefore, if one plans to use Few-Shot Inference in the future, it may be crucial to ensure the correctness of the sample references.**

- **Lastly, incorporating the results after fine-tuning with Qlora, we can see that, compared to training a new language model directly, fine-tuning through adapters for specific application contexts is a more user-friendly and feasible approach, especially for smaller user groups.**

- **Training a language model often requires high computational resources, typically several A100 GPUs, which is not feasible for the average person. On the other hand, using adapters allows us to achieve good fine-tuned model performance at significantly lower costs. Therefore, I believe adapters offer a very promising framework.**