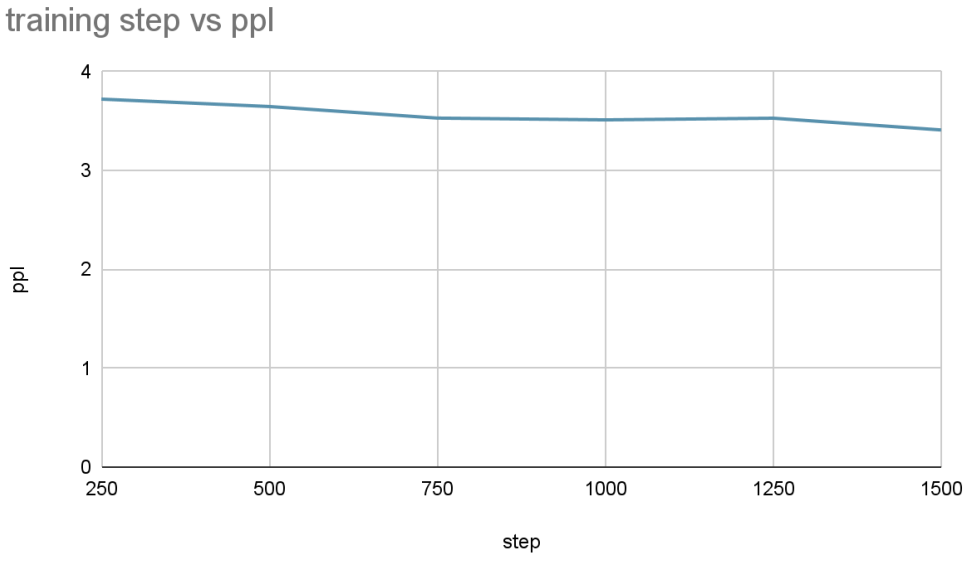
Q1.

Describe

1. How much training data did you use?
   I use 10000 training data to train. A training datum contains id, input, output and instruction. Id, output and instruction column is given by TA, and the input consists of the composition of the instruction and the prompt (from utils.py).
2. How did you tune your model?
   I use qlora.py divided from the TA to tune my model. The trainer is Seq2Seq trainer from huggingface, optimizer is “paged_adamw_32bit”, and the loss function is cross entropy loss. The loss function will use at most 512 length output to count loss.
3. What hyper-parameters did you use?
   Batch Size : 2
   Gradient Accumulation Step : 2
   lora_r : 64
   max_step: 1500
   lr : 2e-4
   Other hyperparameters are as default of qlora.py’s script (https://github.com/artidoro/qlora/blob/main/scripts/finetune_llama2_guanaco_7b.sh).

Performance

1. What is the final performance of your model on the public testing set?
   ppl: 3.4103
   The consequent after inference on the public testing set is OK. Some of them are close to the answer, but some of them are not well. The meaning of the output is roughly OK, however, some 白話文 still use the words from input 文言文, or some 文言文 after translated still use the words from input 白話文.
2. Plot the learning curve on the public testing set



Q2.

Zero-Shot

1. What is your setting? How did you design your prompt? (1%)
   generation strategy:
   do_sample, top_k = 0.6, temperature = 0.6
   prompt :
   ”你是一個古文專家, 今天有用戶想知道以下字句翻譯成文言文或白話文的樣子, 請盡你所能的幫助他, 提供一個完整的答案。”USER:{Instruction}:。\n答案:
   ASSISTANT:”

   I designed my prompt based on the prompt TA had given us. I give it a hint that it is an expert of classical Chinese, which limits its range of answers. We can find out that zero-shot prompting leads to bad results since the model hasn’t been finetuned, it cannot answer the question that well.

Few-Shot (In-context Learning)

1. What is your setting? How did you design your prompt? (1%)
   generation strategy:
   do_sample, top_k = 0.6, temperature = 0.6
   prompt :
   “你是一個古文專家, 今天有用戶想知道以下字句翻譯成文言文或白話文的樣子, 請盡你所能的幫助他, 提供一個完整的答案。舉例而言, 若用戶輸入:「沒過十天, 鮑泉果然被拘捕。\n幫我把這句話翻譯成文言文」則你要回答「後未旬, 果見囚執」, 或者是如果用戶輸入:「翻譯成現代文:\n若使主人少垂古人忠恕之情, 來者側席, 去者剋己, 則僕抗季劄之誌, 不為今日之戰矣。」, 你要回答「如果袁公稍許有點忠恕之情, 來的側席接待, 走的剋己自責, 那我就高季紮逃位之誌, 今天這個仗就不打瞭。」, 或是用戶輸入「宋國派人來祈請山陵, 未獲準許。\n翻譯成文言文:」, 你要回答「宋遣使祈請山陵, 不許。」。讓用戶開始問問題。USER:
   {Instruction}:\n答案: ASSISTANT:”

2. How many in-context examples are utilized? How do you select them? (1%)
   Three in-context examples are utilized, two examples are translation samples from 白話文 to 文言文 and the one left is from 文言文 to 白話文 I decided to use this combination because I think translate 白話文 into 文言文 is more difficult then to translate 文言文 into 白話文. And I’ve also tried adding more in-context examples(up to 5 examples), but the result didn’t have significant improvement. Hence, I choose the three examples described above in my prompt.

Comparison:

3. What’s the difference between the results of zero-shot, few-shot, and LoRA? (2%)

The result of LoRA is the best, and the result of few-shot prompting is a little better than zero-shot prompting. I choose a representative example to illustrate the difference between the three. With the same instruction: “東南水旱, 盜賊常常發生, 西、北二國窺伺日久, 怎麼能不預先計謀？ 於是條列茶鹽、常平等利處與弊端, 於是說: 為今之計, 當其大者, 莫大於土木之功, 現在賜給大臣一棟房子, 有時耗費百萬。幫我把這句話翻譯成文言文。”,
The results:
LoRA: “東南水旱, 盜賊常起, 西北二國窺日久, 豈不先謀？ 於是列茶鹽、常平等利處與弊端, 曰: 為今之計, 當其大者, 莫大於土木之功, 今賜大臣一棟房, 有時耗費百萬。”

zero-shot:”東南無雨, 盜賊常常發生, 西、北二國窺伺日久, 怎麼能不預先計謀？於是條列茶鹽, 常平等利與弊, 說:‘今之計, 當其大者, 莫大於修築城池, 此時賜大臣一屋, 每年花費千金。”

few-shot: “東南無雨, 盜賊多出手, 西、北二國密觀, 何以不事先謀劃？故條列茶鹽, 常平等利與弊, 說:‘今之計, 當其大者, 莫大於修築城池, 此時賜大臣一屋, 每年花費千金。”

We can observe that LoRA is the one which is closest to 文言文. Zero shot prompting do a little work on it but it still looks like 白話文. Compared with few-shot, we can find out that few-shot is closer to 文言文. For Example, in zero-shot, “窺伺日久” is directed copied from user’s instruction, but few-shot’s result turns it into “密觀”. Also, “怎麼能” in the instruction is the same as zero-shot, but few-shot change it into “何以不”, which is more like a 文言文. Overall, LoRA is the best one of the three of results since it is finetuned using the training data on this specific task. LoRA’s result is very close to what we want. Few-shot prompting and Zero-shot prompting’s result didn’t change the instruction so much. But comparing the two, since few shot prompting have some examples to reference to, it can turn some words into 文言文 form, but overall the result is more like 白話文. However, zero-shot prompting can turn very little words into 文言文 form, It looks like the original 白話文 given in the instruction
In other instructions (no matter it is 文言文 translating into 白話文 or 白話文 translating into 文言文), I can find out the similar difference as above.

Q3.

1. Choose one of the following tasks for implementation
   - Experiments with different PLMs
   - Experiments with different LLM tuning methods

I use another LLM tuning method – P-tuning to tune my model.
P-tuning is a tuning method for finding and optimizing for the best prompt representation to the model. It creates a small encoder network for my prompt, tokenizes my prompt and trains on the small encoder network and the prompt. The original network remains the same.

reference: 1. https://www.mercity.ai/blog-post/fine-tuning-llms-using-peft-and-lora
2. https://huggingface.co/docs/peft/task_guides/ptuning-seq-classification

2. Describe your experimental settings and compare the results to those obtained from your original methods

The parameters remain the same (max_step set 250 due to the lack of GPU on colab), the only difference is changing the LoRA config into the p-tuning one.

peft_config = PromptEncoderConfig(task_type="CASUAL_LLM", num_virtual_tokens=20, encoder_hidden_size=128)

Compared with the result of QLoRA, I find out that the P-tuning model is worse than the QLoRA one. Two models are trained with the same parameter, and both are trained under 250 steps, however, the P-tuning model cannot do translation very well. In most cases, it just changes few words and then outputs them. For example, the instruction is “翻譯成文言文: 現在學者大儒，都各自年事已高，教導訓誡的方法，誰來繼承呢？”, and the output of the P-tuning one is “儒者年事已高，誰來繼承呢？”. Though it simplifies the words, where 文言文 often does, the words used are still 白話文’s words. However, the output of the QLoRA one is “ 今學者，各年事已高，誰繼之？”, and I think this is more like the words which exist in 文言文.