ADL Homework 3: Taiwan-LLaMa Instruction Tuning

Department: CSIE   Student ID: d11922016   Name: Jia-Wei, Liao

November 29, 2023

1 LLM Tuning

1.1 Describe

- How much training data did you use?
- How did you tune your model?
- What hyper-parameters did you use?

Solution.

We implement instruction tuning and employ a specific template setting within the prompt. The corresponding input, output, and label structures are illustrated in Figure 1. Within this setting, the token identified as '1' is assigned as the beginning-of-sentence token ID, whereas the token '2' is designated as the end-of-sentence token ID. Additionally, the token with the value '-100' is employed as a padding token, concealing the prompt in the resultant output sentence.

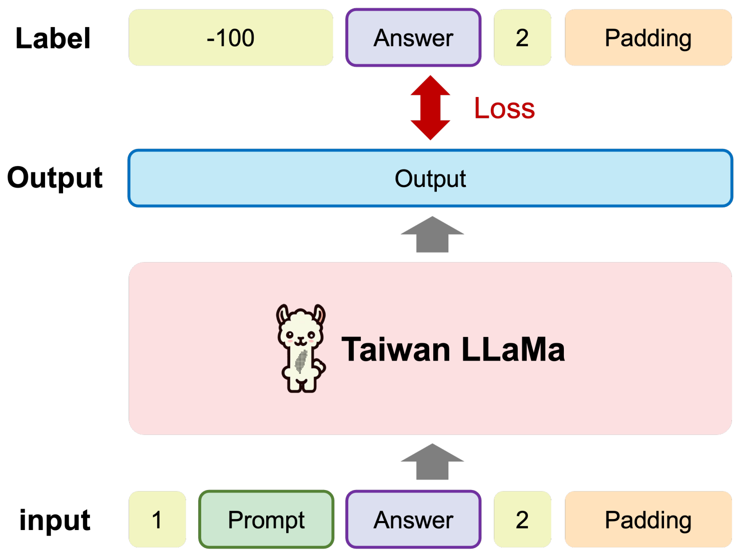
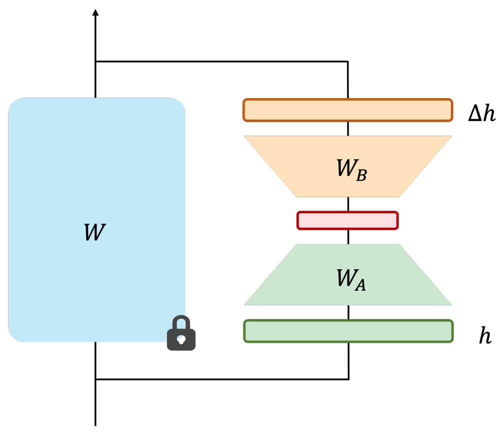
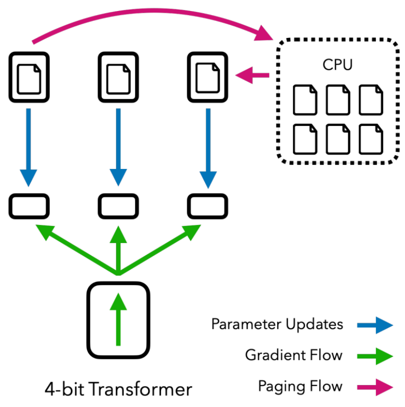


Figure 1: Instruction tuning.

The training dataset comprises 10,000 entries, from which we randomly selected 4,000 for training the model. Since the Taiwan-LLaMa [1] has an large number of parameters, we employed the QLoRA technique [2, 3] to mitigate GPU memory usage during the training phase. The architectural details of both LoRA and QLoRA are displayed in Figure 2. Parameters specifically configured for LoRA include an alpha value set at 16, a dropout rate of 0.1, and a LoRA rank of 16.



(a) LoRA

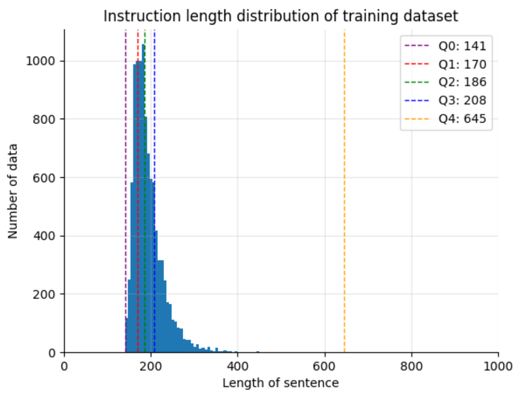


(b) QLoRA

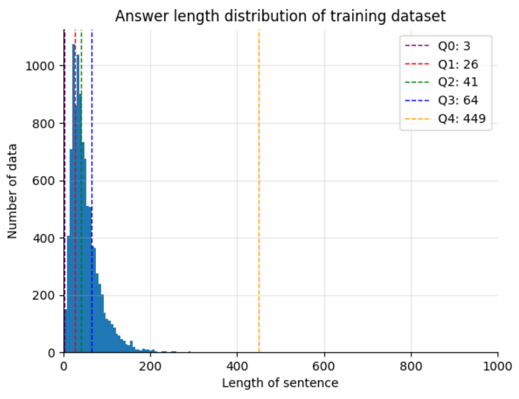
Figure 2: Architecture.

Before commencing the training phase, we conducted an exploratory data analysis with a focus on sentence length distributions. These distributions are presented in Figure 3. The analysis indicated that the majority of sentences across both the training and public datasets typically had a combined length of prompts and answers not exceeding 500 characters. Consequently, in an effort to expedite the training process and enhance memory efficiency, we set the maximum input length to 512.

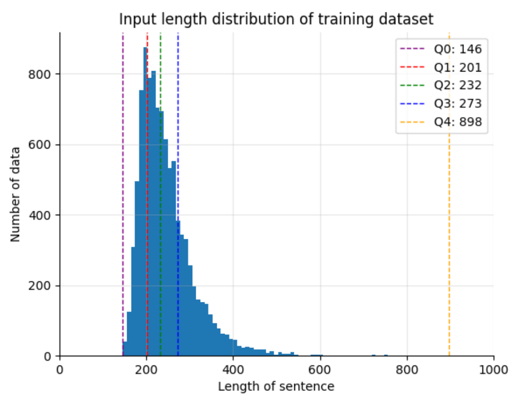
For the training configuration, we established a regimen of 5 epochs with a batch size of 16. and utilized cross-entropy loss function and the AdamW optimizer for parameter optimization. Based on our experimental findings, a learning rate of 2e-4 with constant scheduler demonstrate the optimal performance.



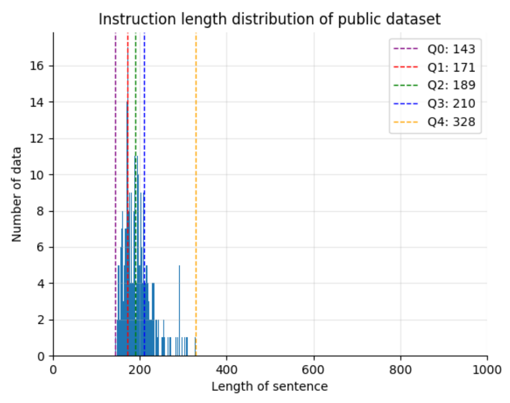
(a) Instruction lengths of training dataset



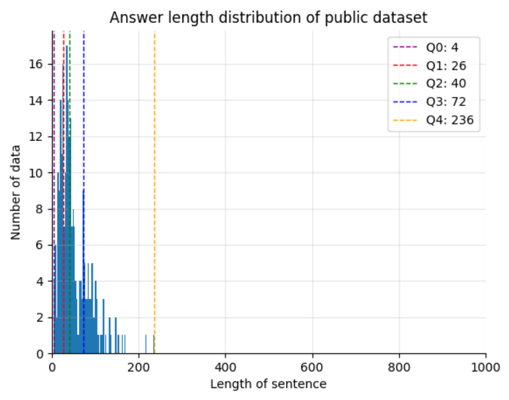
(b) Answer lengths of training dataset



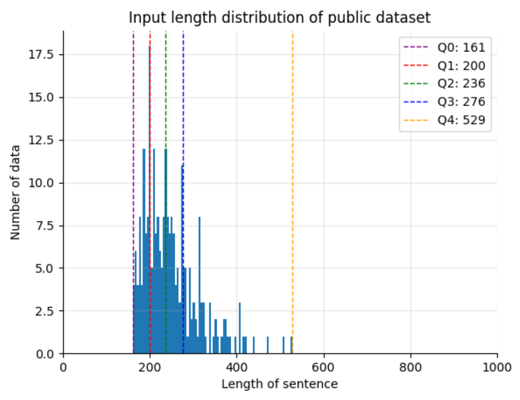
(c) Input lengths of training dataset



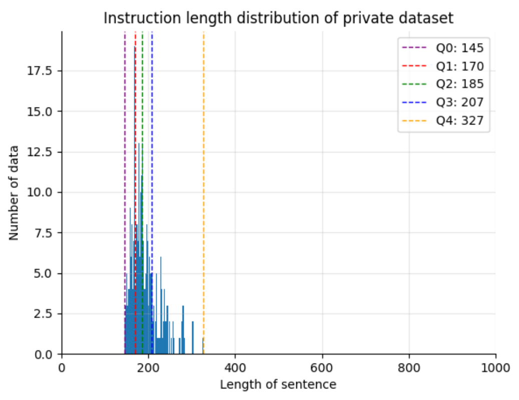
(d) Instruction lengths of public dataset



(e) Answer lengths of public dataset



(f) Input lengths of public dataset



(g) Instruction lengths of private dataset.

Figure 3: Length distribution of training, public and private datasets.

1.2 Show your performance

- What is the final performance of your model on the public testing set?
- Plot the learning curve on the public testing set.

Solution.

In this homework, we utilized perplexity as a metric to evaluate model performance. Perplexity quantifies the degree of confusion of a language model, with lower values indicating better performance. It is defined as follows:

$$PP(\langle w_1, ..., w_n \rangle) = P(w_1, ..., w_n)^{-\frac{1}{n}} = 2^{-\frac{1}{n}\log_2 P(w_1,...,w_n)} \tag{1}$$

The final performance of our model on the public set was a mean perplexity of 3.6493. The learning curve of public set is displayed in Figure 4. The model achieved the best performance at the 1000th step. Due to the implementation of a constant learning rate scheduler, subsequent steps led to overfitting, which resulted in an increase in perplexity.

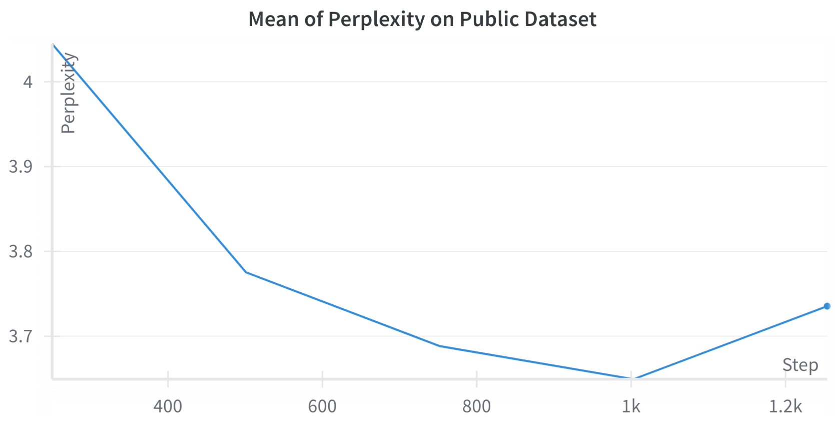


Figure 4: Learning curve on the public testing set.

2 LLM Inference Strategies

2.1 Zero-Shot

What is your setting? How did you design your prompt?

Solution.

Template 1:

The zero-shot learning approach applied to the original prompt resulted in a mean perplexity of 5.4628. The prompt used is as follows:

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。USER: {instruction} ASSISTANT:

Template 2:

We have integrated the following statement into the original prompt: ” 以下的問題爲文言文翻譯成白話文或白話文翻譯成文言文。”. The mean of perplexity improved to 5.2045, a decrease of 0.2583. The prompt is as follows:

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。以下的問題爲文言文翻譯成白話文或白話文翻譯成文言文。 USER: {instruction} ASSISTANT:

Template 3:

We incorporated the role of a high school Chinese language teacher into the prompt, which further reduced the mean of perplexity to 5.1537. The prompt is as follows:

> 你是一名高中國文老師，專門研究文言文與白話文翻譯。現在你要對用戶的問題提供有用、安全、詳細和禮貌的回答。以下的問題爲文言文翻譯成白話文或白話文翻譯成文言文。USER: {instruction} ASSISTANT:

2.2 Few-Shot (In-context Learning)

- What is your setting? How did you design your prompt?
- How many in-context examples are utilized? How you select them?

Solution.

We randomly selected two samples to serve as in-context examples from the training dataset, then incorporated them into the original prompt. We have mean of perplexity of 4.8653.

Template 4:

> 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。提供你兩個例子參考: 1. USER: 翻譯成文言文：雅裏惱怒地說：從前在福山田獵時，你誣陷獵官，現在又說這種話。ASSISTANT: 雅裏怒曰：昔畋於福山，卿誣獵官，今復有此言。2. USER: 辛未，命吳堅爲左丞相兼樞密使，常楙參知政事。把這句話翻譯成白話文。ASSISTANT: 初五，命令吳堅爲左承相兼樞密使，常增爲參知政事。 以下的問題爲文言文翻譯成白話文或白話文翻譯成文言文。請回答: USER: {instruction} ASSISTANT:

Template 5:

We attempt to compute the perplexity scores of the training data using the few-shot method. We visualize the perplexity distribution in Figure 5. We then selected samples with the highest perplexity from both classical Chinese to modern vernacular translations and modern vernacular to classical Chinese translations, and incorporated these into the prompt, which is shown in Table 1. Finally, this approach yielded a mean perplexity of 4.8244 on the public dataset. The prompt is as follows:

> 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。提供你兩個例子參考: 1. USER: 並以個人名義將這件事上奏給皇上。這句話在中國古代怎麼說：ASSISTANT: 通判登州。2. USER: 父往矣。翻譯成白話文：ASSISTANT: 老人傢，按卦上說你早已經死瞭。 以下的問題爲文言文翻譯成白話文或白話文翻譯成文言文。請回答: USER: {instruction} ASSISTANT:

USER	ASSISTANT	Perplexity
並以個人名義將這件事上奏給皇上。這句話在中國古代怎麼說	通判登州	93.6887
父往矣。翻譯成白話文	老人傢，按卦上說你早已經死瞭	31.0981

Table 1: Highest perplexity training samples.

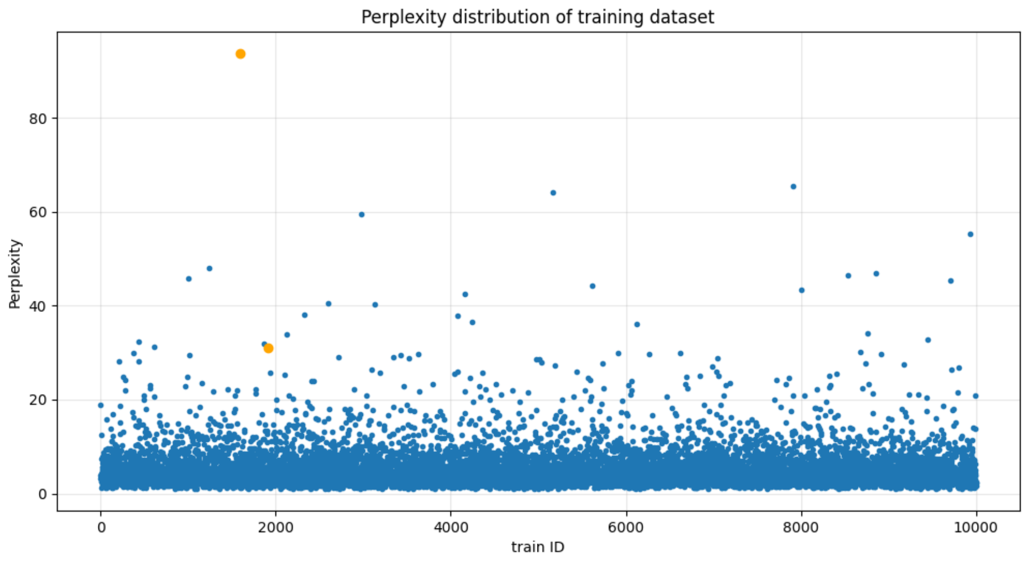


Figure 5: Perplexity distribution of training dataset with few-shot method.

2.3 Comparison

What's the difference between the results of zero-shot, few-shot, and LoRA?

Solution.

From Table 2, it is evident that the results of LoRA fine-tuning performed the best, with the most stable performance in perplexity score. Without training, adopting incontext learning improved perplexity by 0.3392 compared to using zero-shot directly. From Figure 6, it can be observed that after LoRA fine-tuning, the number of outliers in the perplexity distribution significantly decreased. We display the generations of different methods in Table **??**.

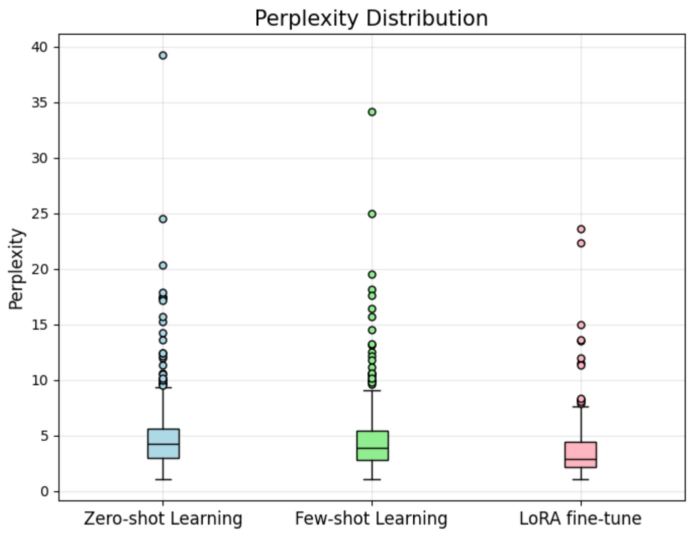


Figure 6: Perplexity distribution of three methods.

Method	Prompt	Mean of Perplexity ↓ (± SD)
Zero-shot Learning	Template 2	5.2045 ± 4.1271
Few-shot Learning	Template 4	4.8653 ± 3.8037
LoRA fine-tuning	Template 2	**3.6476 ± 2.7748**

Table 2: Evaluation results for different methods.

USER	地勢高而乾燥，他們築土爲城牆，架木建房子，用土覆蓋在房屋上麵。
ASSISTANT	其地高燥，築土爲城，架木爲屋，土覆其上。

Table 3: Public data ID: `7192faf3-bd87-4489-9df7-593f410fca90`.

Method	Generation
Zero-shot Learning	地勢高而乾燥，他們築土爲城牆，架木建房子，用土覆蓋在房屋上麵。
Few-shot Learning	地勢高而乾燥，他們築土爲城牆，架木建房子，用土覆蓋在房屋上麵。
LoRA fine-tuning	地高而乾，築土爲城，架木建房，土覆於其上。

Table 4: Generations of different methods for public data.

3 Bonus: Other methods

1. Choose one of the following tasks for implementation
   - Experiments with different PLMs
   - Experiments with different LLM tuning methods
2. Describe your experimental settings and compare the results to those obtained from your original methods

Solution.

We employed the Atom-7B model for a comparative analysis with Taiwan-LLaMa. Atom, an advancement over Llama2 [4], was pre-trained using extensive online Chinese-language data. This model notably expands the coverage of the Chinese character set and includes emojis. Such enhancements make it more efficient in generating texts with emotive expressions. The experimental setup followed the same parameters as outlined in 1. Figure 7 display the training loss and public mean perplexity scores throughout the learning curve. Although Atom-7B exhibits a lower training loss compared to Taiwan-LLaMa, the latter demonstrates superior performance on public mean perplexity. We show the final result in Table 5, Taiwan-LLaMa perform the state-of-the-art result in all the methods.

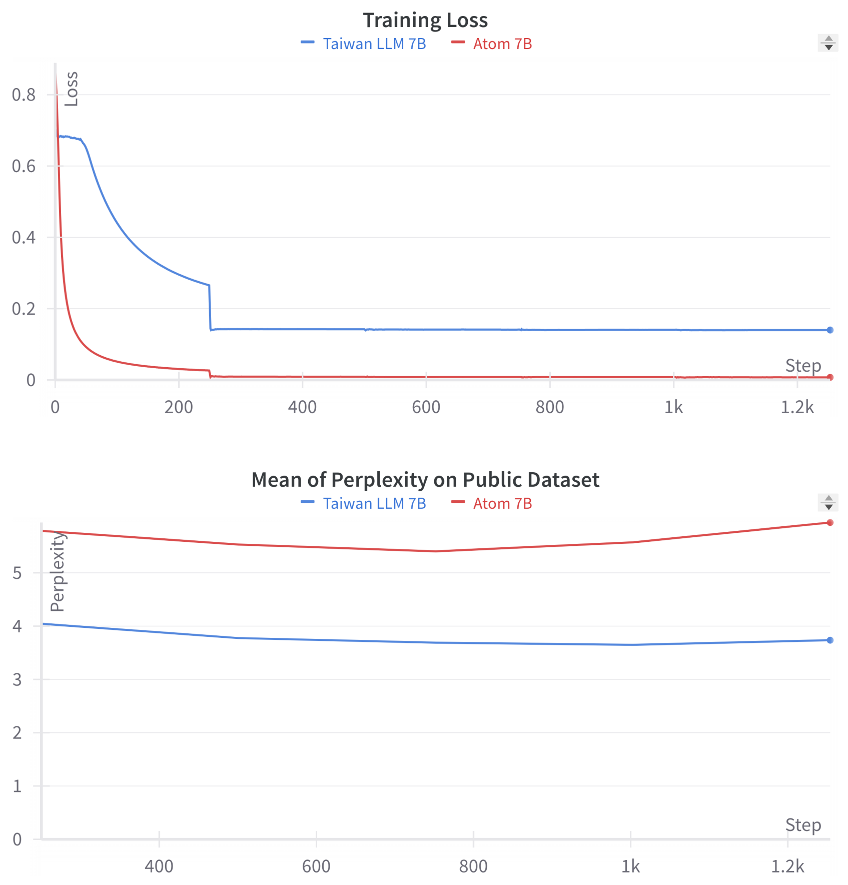


Figure 7: Learning curve of Taiwan-LLaMa and Atom-7B.

Model	Method	Prompt	Mean of Perplexity ↓ (± SD)
Taiwan-LLaMa-7B	Zero-shot Learning	Template 2	**5.2045 ± 4.1271**
Atom-7B	Zero-shot Learning	Template 2	11.4079 ± 17.6943
Taiwan-LLaMa-7B	Few-shot Learning	Template 4	**4.8653 ± 3.8037**
Atom-7B	Few-shot Learning	Template 4	9.1420 ± 9.0381
Taiwan-LLaMa-7B	LoRA fine-tuning	Template 2	**3.6476 ± 2.7748**
Atom-7B	LoRA fine-tuning	Template 2	5.6042 ± 5.6114

Table 5: Evaluation results for different methods.

References

[1] Yen-Ting Lin and Yun-Nung Chen. Language models for taiwanese culture, 2023. Code and models available at https://github.com/MiuLab/Taiwan-LLaMa.

[2] Edward J Hu, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. Lora: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2021.

[3] Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. Qlora: Efficient finetuning of quantized llms. *arXiv preprint arXiv:2305.14314*, 2023.

[4] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.

Appendix

All training sessions were conducted on a PC running Ubuntu 20.04.2 with a 12th Gen Intel(R) Core(TM) i7-12700 CPU, equipped with a single NVIDIA GeForce RTX 4090 GPU featuring 24 GB of dedicated memory.

Environment

To set the environment, you can run this command:

```
pip install -r configs/requirements.txt
```

Download

To download the datasets and model checkpoint, you can run the command:

```
bash ./download.sh
```

Reproducing

To reproduce the result of inference, you can run the command:

```
bash ./run.sh \
    <pretrain_model_folder> \
    <lora_model_folder> \
    <input_data_path> \
    <output_file_path>
```

Training

To fine-tune the Taiwan-LLaMa model, you can run the command:

```
python train.py \
       --base_model_path <pretrain model folder> \
       --train_data_path <train data path> \
       --valid_data_path <valid data path> \
       --train_num <number of used training data> \
       --epoch <number of epochs> \
       --batch_size <number of training batch size> \
       --accum_grad_step <number of accumulated gradient batch size> \
       --lr <learning rate> \
       --lr_scheduler <learning rate scheduler> \
       --warm_up_step <number of warm up step>
       --lora_rank <rank of LoRA>
```

3.1 Inference

To inference the Taiwan-LLaMa model, you can run the command:

```
python infer.py \
    --method <support method: lora-fine-tune, zero-shot, and few-shot> \
    --base_model_path <pretrain model folder> \
    --peft_path <lora model folder> \
    --test_data_path <test data path> \
    --output_path <output file oath>
```

3.2 Demo

To demo the conversation with Taiwan-LLaMa model, you can run the command:

```
python demo.py \
    --method <support method: lora-fine-tune, zero-shot, and few-shot>
   --base_model_path <pretrain model folder> \
   --peft_path <lora model folder> \
   --test_data_path <test data path>
```