ADL-HW3

Q1: LLM Tuning

Describe:

How much training data did you use? (2%)

I limited the training data to a maximum of 5,000 samples (--max_train_samples 5000) from the dataset located at hw3/data/train.json.

How did you tune your model? (2%)

The tuning was performed using a few key configurations:

LoRA (Low-Rank Adaptation): LoRA is applied with parameters specific to low-rank adaptation, allowing fine-tuning with fewer parameters while preserving model efficiency.

Mixed Precision (bfloat16): The --bf16 flag enables mixed precision with bfloat16, which reduces memory usage and speeds up training on compatible hardware.

Gradient Accumulation: Since the batch size was not directly specified, it's likely simulated using gradient accumulation (from the original script) to effectively handle the memory load.

Saving Checkpoints: Checkpoints are saved every 30 steps (--save_steps 30) to allow for recovery and monitoring of training progress.

What hyper-parameters did you use? (2%)

Model: zake7749/gemma-2-2b-it-chinese-kyara-dpo

Learning Rate: Set in the script as 0.0002

LoRA Parameters:

lora_r=64: Low-rank adaptation dimension.

lora_alpha=16: Scaling factor for LoRA.

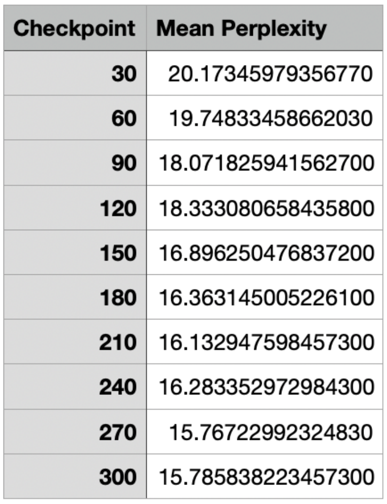
lora_dropout=0.0: No dropout applied.

Max Steps: 300 steps (--max_steps 300) to limit the training duration.

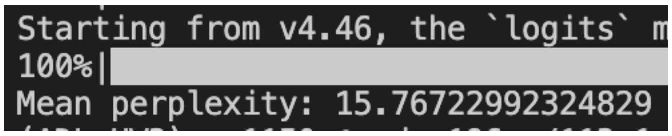
Precision: bf16 mixed precision to reduce memory usage and speed up computation.

Performance:

The table below shows the mean perplexity values at each checkpoint, rounded to three decimal places for clarity.

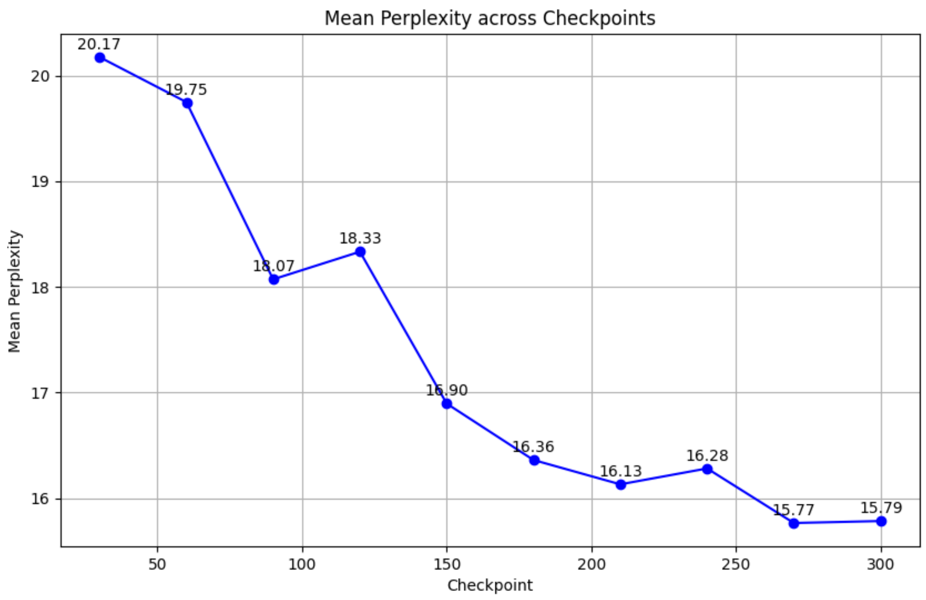


Checkpoint	Mean Perplexity
30	20.17345979356770
60	19.74833458662030
90	18.071825941562700
120	18.333080658435800
150	16.896250476837200
180	16.363145005226100
210	16.132947598457300
240	16.283352972984300
270	15.76722992324830
300	15.785838223457300



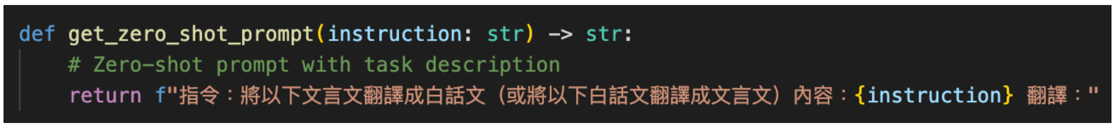
```
Starting from v4.46, the `logits` m
100%|
Mean perplexity: 15.76722992324829
```

The learning curve plot with rounded mean perplexity values highlights a consistent decrease in perplexity across checkpoints, demonstrating the model's improved performance as training progresses.



Q2: LLM Inference Strategies

Zero-Shot:



```python
def get_zero_shot_prompt(instruction: str) -> str:
    # Zero-shot prompt with task description
    return f"指令：將以下文言文翻譯成白話文（或將以下白話文翻譯成文言文）內容：{instruction} 翻譯："
```

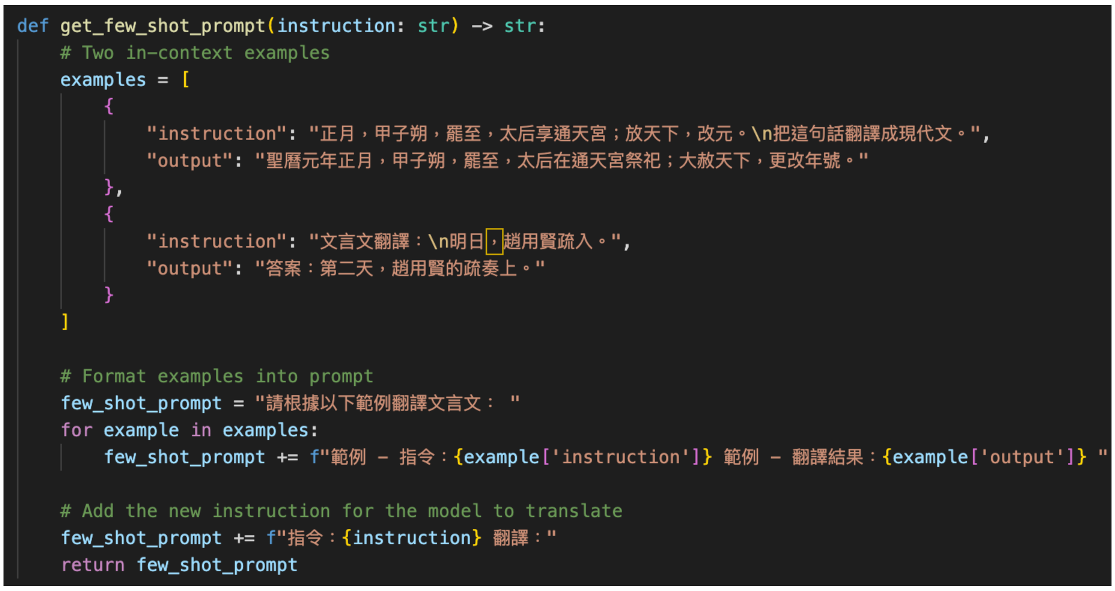
Setting:

Using the get_zero_shot_prompt function to generate the prompt, which includes only a basic instruction directing the model to translate the given text into either vernacular Chinese or Classical Chinese. For example, "Instruction: Translate the following Classical Chinese text into vernacular Chinese (or vice versa)." The prompt includes the specific text to be translated.

Prompt Design:

The prompt solely provides a translation directive without any examples, relying on the model's inherent knowledge to complete the translation.

Few-Shot:



```python
def get_few_shot_prompt(instruction: str) -> str:
    # Two in-context examples
    examples = [
        {
            "instruction": "正月，甲子朔，罷至，太后享通天宮；放天下，改元。\n把這句話翻譯成現代文。",
            "output": "聖曆元年正月，甲子朔，罷至，太后在通天宮祭祀；大赦天下，更改年號。"
        },
        {
            "instruction": "文言文翻譯：\n明日，趙用賢疏入。",
            "output": "答案：第二天，趙用賢的疏奏上。"
        }
    ]

    # Format examples into prompt
    few_shot_prompt = "請根據以下範例翻譯文言文： "
    for example in examples:
        few_shot_prompt += f"範例 - 指令：{example['instruction']} 範例 - 翻譯結果：{example['output']} "

    # Add the new instruction for the model to translate
    few_shot_prompt += f"指令：{instruction} 翻譯："
    return few_shot_prompt
```

Setting:

Using the get_few_shot_prompt function to generate the prompt, which includes two examples as in-context references to help the model better understand the translation task. Each example includes an instruction and the expected translation outcome for the model to refer to.

Prompt Design:

The prompt contains multiple examples of "instruction" and "translation result" pairs, allowing the model to use these as context. The new instruction for the model to translate is appended at the end.

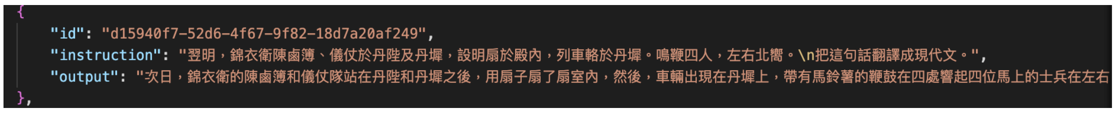
Comparison:

Zero-Shot:

Zero-Shot refers to a scenario where the model performs the task without any examples provided. The model relies solely on the instruction to infer a response, without any contextual examples to aid in understanding the specifics of the task. Zero-Shot is suitable for straightforward instructions and predictions, but it may lack depth when handling more complex tasks.

Pros: Quick execution, no need for extra data preparation.

Cons: May struggle with nuanced context, lacking accuracy and detail.



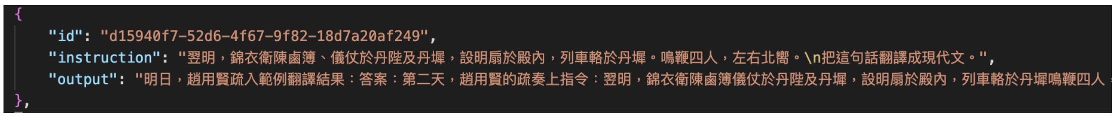
```
{
    "id": "d15940f7-52d6-4f67-9f82-18d7a20af249",
    "instruction": "翌明，錦衣衛陳鹵簿、儀仗於丹陛及丹墀，設明扇於殿內，列車輅於丹墀。鳴鞭四人，左右北嚮。\n把這句話翻譯成現代文。",
    "output": "次日，錦衣衛的陳鹵簿和儀仗隊站在丹陛和丹墀之後，用扇子扇了扇室內，然後，車輛出現在丹墀上，帶有馬鈴薯的鞭鼓在四處響起四位馬上的士兵在左右
},
```

Few-Shot:

Few-Shot means providing the model with a few examples before it performs the task, helping it understand the expected format and style. These examples guide the model on how to respond to the instruction, resulting in outputs that are more aligned with expectations. Few-Shot is useful for scenarios requiring context understanding and can yield more human-like and richer responses.

Pros: A small number of examples can significantly improve accuracy, and the generated content better matches expectations.

Cons: Requires preparation of example data and may increase processing time.



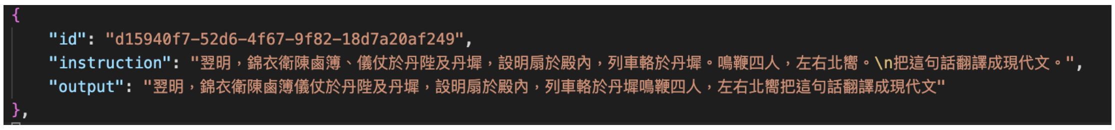
```
{
    "id": "d15940f7-52d6-4f67-9f82-18d7a20af249",
    "instruction": "翌明，錦衣衛陳鹵簿、儀仗於丹陛及丹墀，設明扇於殿內，列車輅於丹墀。鳴鞭四人，左右北嚮。\n把這句話翻譯成現代文。",
    "output": "明日，趙用賢疏入範例翻譯結果：答案：第二天，趙用賢的疏奏上指令：翌明，錦衣衛陳鹵簿儀仗於丹陛及丹墀，設明扇於殿內，列車輅於丹墀鳴鞭四人
},
```

LoRA:

LoRA is a fine-tuning technique that adapts the model for specific tasks by using a small number of additional parameters. It modifies only a subset of the original model parameters, enabling it to achieve high accuracy for specific tasks with limited computational resources. LoRA is effective for preserving the model’s original knowledge while also adapting it to new tasks.

Pros: Efficient and resource-friendly, ideal for improving performance on specific tasks without altering the entire model.

Cons: In some cases, may not perform as well as full fine-tuning.



```
{
    "id": "d15940f7-52d6-4f67-9f82-18d7a20af249",
    "instruction": "翌明，錦衣衛陳鹵簿、儀仗於丹陛及丹墀，設明扇於殿內，列車輅於丹墀。鳴鞭四人，左右北嚮。\n把這句話翻譯成現代文。",
    "output": "翌明，錦衣衛陳鹵簿儀仗於丹陛及丹墀，設明扇於殿內，列車輅於丹墀鳴鞭四人，左右北嚮把這句話翻譯成現代文"
},
```

Q3: Bonus: Llama3-Taiwan 8B

Strategies:

Experimental Setup

Model: The main change was switching from the original model zake7749/gemma-2-2b-it-chinese-kyara-dpo to Llama3-Taiwan (8B), which is specifically trained on traditional Chinese data.

Parameters: All other parameters, such as learning rate (0.0002), LoRA settings (lora_r=64, lora_alpha=16, lora_dropout=0.0), batch size, and max_steps=300, were kept the same to ensure a fair comparison.

Data: Both models were fine-tuned on the same classical Chinese dataset.

Results Comparison

Perplexity Trends:

Kyara Model: The original model started with a higher perplexity of approximately 20 and gradually decreased to 15.79 by checkpoint 300. This shows a steady reduction in perplexity as training progressed, indicating improved language model performance over time.

Llama3-Taiwan (8B): This model started with a much lower initial perplexity of around 7.18 and further reduced to 6.10 by checkpoint 300. This low perplexity from the start suggests that Llama3-Taiwan (8B) is better suited for traditional Chinese, likely due to its pretraining on traditional Chinese data.

Performance:

Llama3-Taiwan (8B) consistently maintained lower perplexity values than the original model at every checkpoint. This indicates that it was able to adapt more effectively to the classical Chinese data, demonstrating superior language understanding and generation capabilities for traditional Chinese.

The improved performance can be attributed to Llama3-Taiwan's pretraining on a traditional Chinese dataset, which provides a closer linguistic foundation for classical Chinese text.

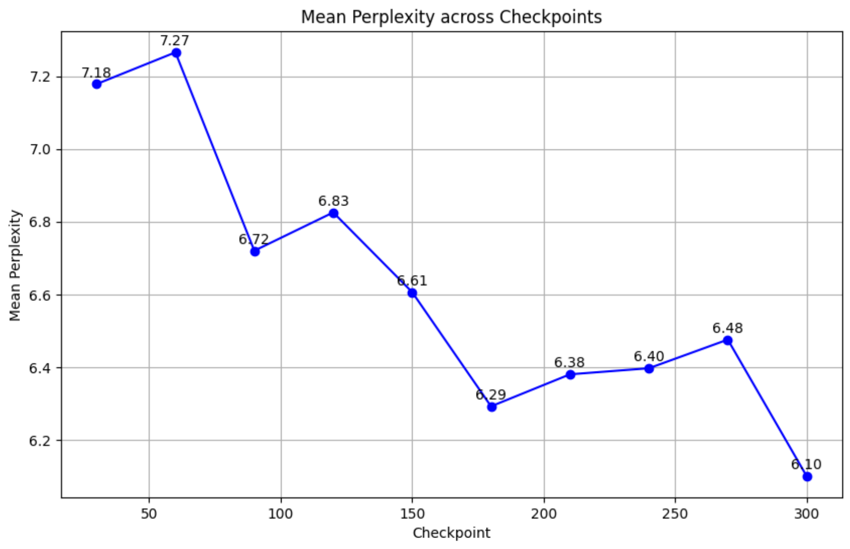
Conclusion

Llama3-Taiwan (8B) proved to be significantly more effective for this task compared to the original model. Its lower perplexity values show that it is inherently better at handling traditional Chinese, making it a preferable choice when working with classical Chinese data.

Using Llama3-Taiwan (8B) for classical Chinese tasks, based on this setup, resulted in a more accurate and contextually appropriate model with faster convergence, as evidenced by the lower perplexity throughout training.

This experiment demonstrates that models pretrained on linguistically similar data can yield better results in fine-tuning tasks, especially when language or dialect-specific nuances are involved.

Checkpoint	Mean Perplexity
30	7.178369957447050
60	7.265611580848690
90	6.720559018611910
120	6.825776010036470
150	6.606883071422580
180	6.293017171382900
210	6.3809881882667500
240	6.397646416187290
270	6.476371746063230
300	6.101068610191350


Mean Perplexity across Checkpoints
Mean Perplexity
Checkpoint