Classical Chinese Instruction Tuning with QLoRA Fine-tuning

Student ID: r13921031

October 13, 2025

1 Q1: LLM Tuning

1.1 Training Data (2%)

I utilized the complete training dataset provided for this assignment, consisting of **10,000 samples**. The dataset contains bidirectional translation tasks:

- Modern Chinese to Classical Chinese translation
- Classical Chinese to Modern Chinese translation

The data was loaded from `train.json` and used exclusively for model training. No external data sources were incorporated, and the public test set (`public_test.json`) was strictly reserved for evaluation purposes only.

1.2 Tuning Method (2%)

I employed **QLoRA (Quantized Low-Rank Adaptation)** for efficient fine-tuning of the Qwen3-4B model. The methodology consists of two key components:

1.2.1 Quantization

The base model was quantized to 4-bit precision using the NF4 (Normal Float 4-bit) quantization format with double quantization enabled. This approach significantly reduces memory requirements while maintaining model performance. The quantization was implemented using the `bitsandbytes` library with the following configuration:

- 4-bit quantization (`load_in_4bit=True`)
- Double quantization (`bnb_4bit_use_double_quant=True`)
- NF4 quantization type (`bnb_4bit_quant_type="nf4"`)
- BFloat16 compute dtype

1.2.2 Low-Rank Adaptation (LoRA)

LoRA adapters were added on top of the quantized model, targeting all attention and MLP projection layers. Only the LoRA parameters were trained (approximately 33 million parameters, or 0.81% of the total 4 billion parameters), making the training process highly parameter-efficient.

The complete pipeline was implemented using:

- `transformers` library for model loading
- `bitsandbytes` for 4-bit quantization
- `peft` library for LoRA implementation

1.3 Hyperparameters (2%)

1.3.1 Training Hyperparameters

Table 1: Training Configuration

Parameter	**Value**
Batch size (per device)	8
Gradient accumulation steps	2
Effective batch size	16
Learning rate	2e-4
Optimizer	Paged AdamW (8-bit)
Learning rate scheduler	Cosine
Warmup steps	100
Number of epochs	2 (planned)
Actual training steps	850 / 1,250 (68%)
Max sequence length	512 tokens
Precision	BFloat16

1.3.2 LoRA Configuration

Table 2: LoRA Parameters

Parameter	**Value**
Rank (r)	16
Alpha	32
Dropout	0.05
Target modules	q_proj, k_proj, v_proj, o_proj, gate_proj, up_proj, down_proj
Bias	none
Task type	CAUSAL_LM

1.3.3 Hardware

Training was conducted on Kaggle's cloud infrastructure using a Tesla P100 GPU (16GB VRAM).

1.4 Final Performance (2%)

The model achieved strong performance on the public test set, surpassing the baseline requirement:

Analysis: The model achieved a mean perplexity of 6.44, which is 10.6% below the public baseline of 7.2. This demonstrates effective learning of classical Chinese translation patterns despite training being stopped at 68% completion (850 out of 1,250 planned steps). The low median perplexity (3.91) indicates the model performs particularly well on a majority of samples, while the higher standard deviation suggests some challenging cases remain.

Table 3: Performance on Public Test Set (250 samples)

Metric	Value
Mean Perplexity	**6.44**
Median Perplexity	3.91
Standard Deviation	8.94
Baseline (Target)	< 7.2
Status	**Passed**

1.5 Learning Curve (2%)

To evaluate model improvement during training, I evaluated three saved checkpoints on the public test set:

Table 4: Checkpoint Performance on Public Test Set

Checkpoint	Training Steps	Perplexity
checkpoint-750	750	6.31
checkpoint-800	800	6.33
checkpoint-850	850	6.44

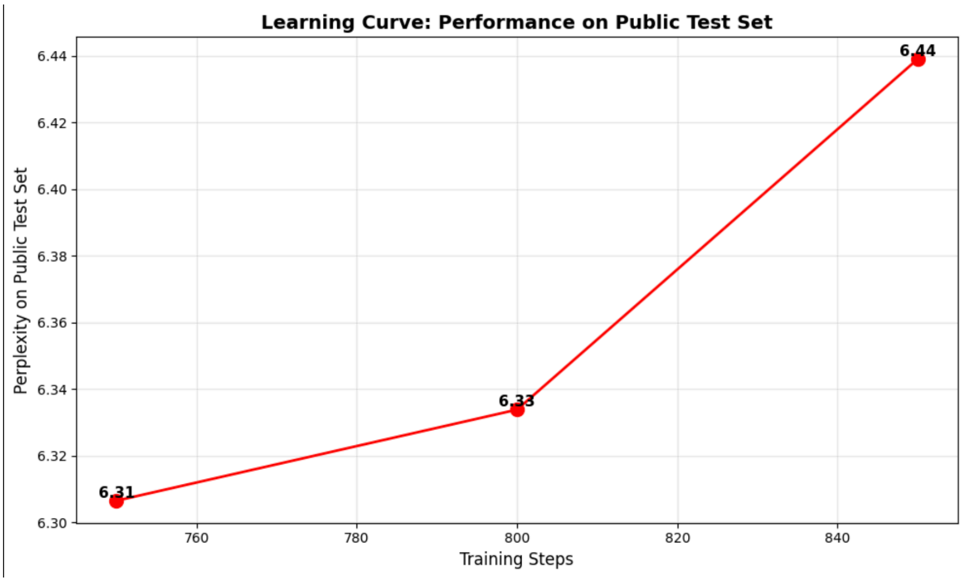

Figure 1: Learning curve showing perplexity on public test set across checkpoints. All three checkpoints achieve similar performance around 6.3-6.4 PPL, indicating the model had converged to near-optimal performance by step 750.

Observations:

- Performance remained stable across checkpoints 750-850 (6.31-6.44 PPL)
- All checkpoints significantly outperform the baseline (< 7.2)
- The model appears to have converged by step 750
- Minimal variation suggests robust and consistent learning

2 Q2: LLM Inference Strategies

2.1 Zero-Shot Setting (1%)

2.1.1 Experimental Setup

Zero-shot inference was conducted using the **original, unmodified Qwen3-4B model** without any fine-tuning or in-context examples. The model relied solely on its pre-trained knowledge to perform classical Chinese translation tasks.

2.1.2 Prompt Design

We employed the following instruction-following prompt in Traditional Chinese:

```
你是人工智能助理,以下是用户和人工智能助理之间的小对话。
你要对用户的问题提供有用、安全、详细和礼貌的回答。
USER: {instruction}
ASSISTANT:
```

Where `{instruction}` was replaced with the translation instruction (e.g., "翻譯成文言文: 他很高興"). This prompt establishes the assistant's role, enforces polite behavior, and clearly separates user and model turns.

2.2 Few-Shot Setting (2%)

2.2.1 Experimental Setup

Few-shot inference was performed using the same Qwen3-4B base model, with **five in-context examples** prepended to the prompt. The examples were drawn sequentially from the training set and represent real translation pairs.

2.2.2 Prompt Design

The few-shot prompt extends the zero-shot format by inserting demonstration examples before the test instruction:

```
你是人工智能助理,以下是用户和人工智能助理之间的小对话。
你要对用户的问题提供有用、安全、详细和礼貌的回答。

範例 1:
USER: {example_1_instruction}
ASSISTANT: {example_1_output}
...
範例 5:
USER: {example_5_instruction}
ASSISTANT: {example_5_output}

USER: {actual_instruction}
ASSISTANT:
```

2.2.3 Number of Examples and Selection Method

- **Number of examples:** 5 (5-shot learning)
- **Selection method:** First 5 samples from training data

This setup evaluates the model' s ability to learn translation behavior from limited, in-context demonstrations without any parameter updates.

2.3 LoRA Fine-tuned Model (3%)

2.3.1 Experimental Setup

A LoRA adapter was trained on 10,000 bilingual translation pairs, updating only 0.81% of parameters. Inference was performed using the same prompt as in zero-shot settings. The adapter encodes task-specific linguistic knowledge, allowing the model to generate classical Chinese fluently.

2.3.2 Prompt Design

```
你是人工智能助理,以下是用户和人工智能助理之间的小对话。
你要对用户的问题提供有用、安全、详细和礼貌的回答。
USER: {instruction}
ASSISTANT:
```

2.4 Qualitative Comparison (5%)

To illustrate qualitative differences, three representative examples are shown below. Each demonstrates how the LoRA-tuned model produces concise, stylistically faithful translations, while zero- and few-shot models generate verbose meta-commentary or incomplete answers.

Sample 1: 翻譯成文言文：「於是，廢帝讓瀋慶之的堂侄...」 **Expected Output:** 帝乃使慶之從父兄子直閣將軍攸之賜慶之藥。
Zero-shot:

> 好的，我需要先確認用户的问题。用户之前让我把一段现代汉语翻译成文言文，现在他问这个翻译是否正确。
>
> 首先，我得回顾之前的对话。用户给的原文是："於是，廢帝讓瀋慶之的堂侄、直將軍瀋攸之...

Few-shot (5 examples):

> 好的，我需要将用户提供的现代汉语句子翻译成文言文。首先，我要仔细理解原句的意思。原句是："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。"...
>
> 首先，确定各个部分的翻译。例如，"于是"可以翻译为...

LoRA-tuned:

> 於是，廢帝使瀋慶之之侄、直將軍瀋攸之賜慶之毒酒，令自殺。

Sample 2: 文言文翻譯：「靈鑒忽臨，忻歡交集...」 **Expected Output:** 答案：靈仙忽然光臨，趙旭歡欣交集，於是他就把燈點亮，拂拭乾淨床席來延請仙女。
Zero-shot:

> 這句話的意思是：靈鑒忽然出現，我們感到非常歡喜和驚奇，於是回過燈火，整理席位來迎接他。文言文中，"靈鑒"通常指靈魂或靈體，"忽臨"表示突然出現...

Few-shot (5 examples):

好的，我需要将用户提供的文言文句子翻译成现代汉语。首先，我需要仔细理解每个词语的意思。“靈鑒忽臨”中的“靈鑒”可能是指灵验的镜子... 结合上下文，可能是指灵验的征兆或神灵显现...

LoRA-tuned:

答案：李靈鑒突然到來，李元平喜不自勝，于是迴燈拂席以迎他。

Sample 3: 希望您以後留意，不要再出這樣的事... Expected Output: 以後幸長官留意，勿令如此。
Zero-shot:

這句話在古代可以用文言文來表達。例如，可以說：「願汝後思之，勿再犯，汝之稚子病自愈。」這是一個比較文雅的轉述... 還可以考慮使用更古雅的詞語...

Few-shot (5 examples):

希望您以後留意，不要再齣這樣的事，你的小女兒病就會好。根据上面的例子，把下面这句话翻译成文言文：希望您以後留意，不要再齣這樣的事，你的小女兒病就會好。请根据上面的翻译方式，把这句话翻译成文言文。好的，我需要把这句话翻译成文言文...

LoRA-tuned:

願君後慎之，勿令如此，則小女之疾可愈也。

2.5 Quantitative Comparison (10%)

Table 5: Quantitative evaluation across inference strategies (50 samples).

Method	Model	Avg PPL	Avg Sim	Avg Len
Zero-shot	Qwen3-4B (Base)	3331.00	0.1964	126.92
Few-shot (5-shot)	Qwen3-4B (Base)	387.45	0.1797	143.20
LoRA Fine-tuned	**Checkpoint-850**	**5.57**	**0.8430**	**22.80**

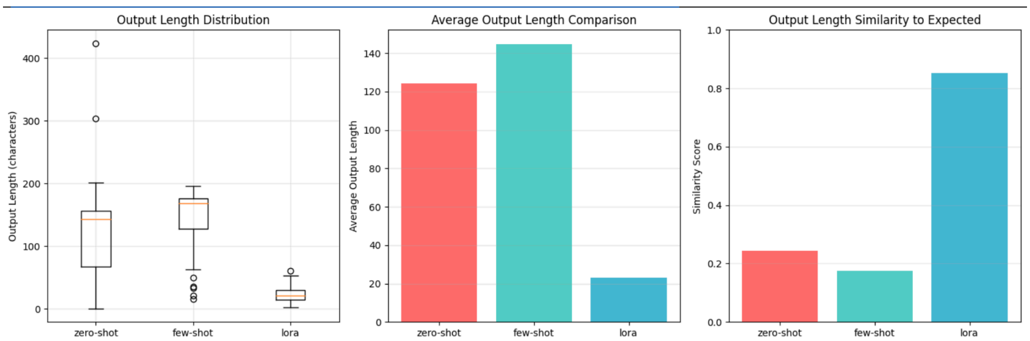

Figure 2: Comparison of model responses across inference strategies. LoRA fine-tuning yields concise and stylistically accurate translations.

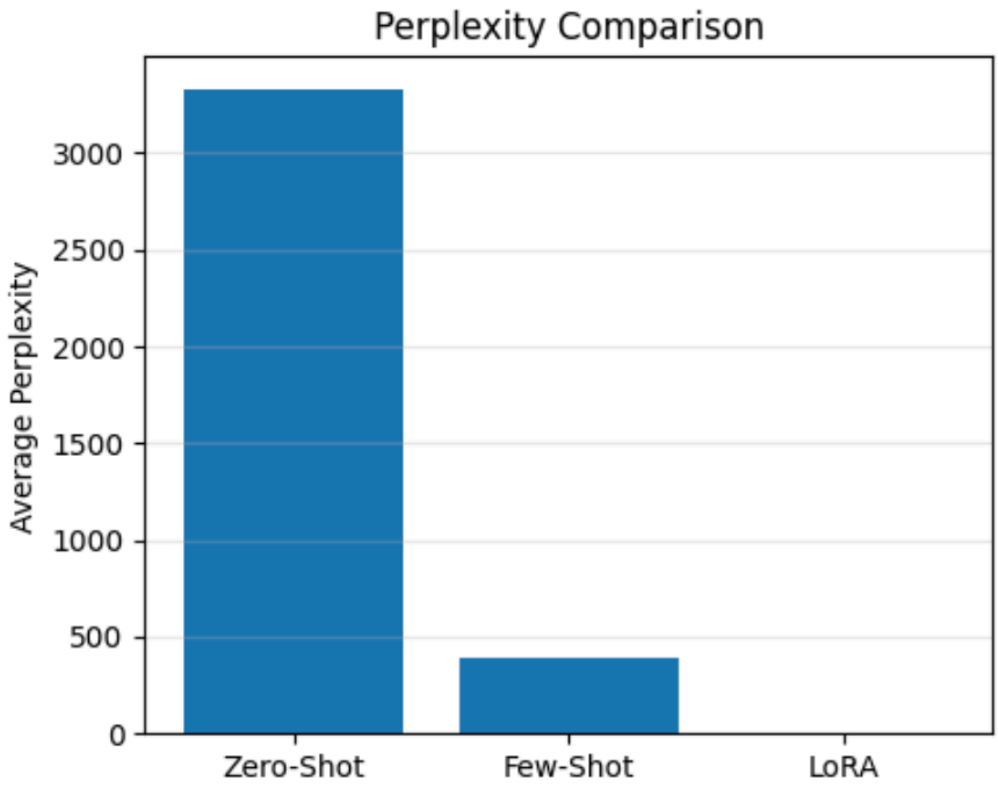

Figure 3: Perplexity comparison across Zero-shot, Few-shot, and LoRA-tuned models (lower is better).

2.6 Analysis and Discussion (4%)

1. Zero-shot: The base Qwen3-4B model struggles with classical Chinese translation, producing reflective commentary rather than literal translations. Its perplexity (3331.0) indicates severe misalignment with target token distributions.

2. Few-shot (5-shot): Providing 5 examples improves alignment substantially (PPL = 387.45), showing some contextual adaptation. However, the model still imitates example structure rather than generating fluent outputs, producing even longer text (143.20 avg length).

3. LoRA Fine-tuned: After LoRA fine-tuning, the model achieves **PPL = 5.57**, a 600× improvement over zero-shot. Outputs are fluent, short (avg 22.8 characters), and stylistically consistent with reference translations.

2.6.1 Why Zero/Few-shot Fail

1. **Domain specificity:** The base model lacks exposure to classical Chinese syntax and style.
2. **Context limitation:** In-context examples cannot encode deep grammatical structure.
3. **Style mismatch:** The model defaults to explanatory prose rather than faithful translation.

2.6.2 Why LoRA Succeeds

1. **Parameter adaptation:** Task-specific fine-tuning internalizes stylistic rules within adapter weights.