NLP Final Project

Default Project / RAG Improvement via LoRA Fine-Tuning

2025.06.09

KangJun Lee

Introduction

- In the default project…

Example Output (from summary model)

```
Indiana town's Memories Pizza is
shut down after online threat....
(omission until end)
and and and and and and...
```

Example 1: Sample summary output

Metric	Score
ROUGE-1	0.0025
ROUGE-2	0.0000
ROUGE-L	0.0025
ROUGE-Lsum	0.0025

Table 1: ROUGE scores for the summarization task after fine-tuning

Problem)

GPT-small model in Default Project has strange output such as repeating words "and and and …" in the end of sentence. → Leads to low score in GPT model and RAG finetuned model.

Idea : How about improving quality of output of GPT by finetuning?

Approach

Method : Use finetune.py in Self-RAG [1)]

In Self-RAG method, there is step for finetuning model with **Instruction-following generation** task, using LoRA.

Used the same finetune.py script with minor modifications to fit our model configuration.

Dataset

Use same dataset as in Self-RAG finetuning.

Dataset consists of a collection of available sources, including FLAN v2, ARC _Easy, NQ, and others.

For fast training, randomly select 30% of data (45,000 instructions-output pair) from original dataset.

(In finetune.py, LoRA is implemented by PEFT library.)

```
 Instruction:\n Q: Is there a negative
or positive tone to this product review?
...
Response:\n
[No Retrieval]Negative[Utility:5]
```

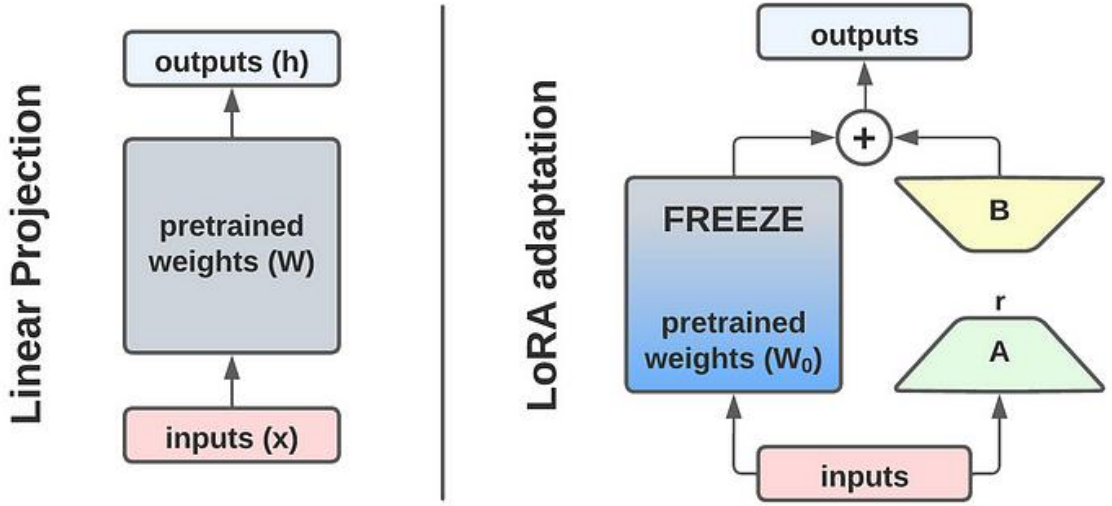
(example of dataset)

1) Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, andHannaneh Hajishirzi. 2023. Self-rag: Learning to re-trieve, generate, and critique through self-reflection.In Proceedings of the Twelfth International Confer-ence on Learning Representations (ICLR).

Approach

Idea : Use LoRA method to Finetune GPT model.

Q. What is LoRA(Low Rank Adaptation)?



LoRA freezes the original weights (W_0) and introduces two small trainable matrices: A and B.

Update, $W = W_0 + BAx$

$(A \in \mathbb{R}^{d \times r}, B \in \mathbb{R}^{r \times d})$

Advantage : Only small number of parameters (A and B) are updated, so can reduce time.

Extra experiment

1. Apply LoRA finetuning to summary Task.

Compare original summary model and LoRA model with ROGUE score.

2. Apply LoRA finetuning to RAG model.

Compare original RAG model and LoRA finetuned model with ROGUE score and accuracy.

3. Apply LoRA finetuning to GPT model, and apply RAG finetuning. (LoRA_RAG model)

Compare original RAG model and RAG model of LoRA finetuned GPT model.

Results

1-1. Summary task & Classification task.

- ROUGE and classification accuracy were lower than expected.
- In my opinion, it is caused by the overfitting due to small parameter size and dataset.
- Token_Acc = 99.9% and Perplexity = 1.73 during Pretraining
- Strange repetition of words in output.

Metric	Score
ROUGE-1	0.0025
ROUGE-2	0.0000
ROUGE-L	0.0025
ROUGE-Lsum	0.0025

Table 1: ROUGE scores for the summarization task after fine-tuning

Metric	Score
Accuracy	0.3818

Table 2: Classification accuracy of the finetuned-GPT model for CF

Example Output (from summary model)

```
Indiana town's Memories Pizza is
shut down after online threat....
(omission until end)
and and and and and and...
```

Example 1: Sample summary output

Results

1-2. RAG model and Zero-shot RAG

- ROUGE scores were significantly lower than zero-shot RAG with LLaMA-3.2B-Instruct.
- While some gap is expected due to LLaMA's large model size, we believe the main cause is overfitting in our small pretrained model.

Metric	A	B
Accuracy	0.0002	0.0243
ROUGE-1	0.0006	0.0313
ROUGE-2	0.0000	0.0069
ROUGE-L	0.0006	0.0311
ROUGE-Lsum	0.0007	0.0313

Table 3: Performance comparison between RAG(A) and zero-shot RAG using LLaMA-3.2B-Instruct(B).

Results

1-3. Changing Prompt in Zero-Shot RAG.

Original Prompt

```
Title:
Passage: {passage_1}
Title:
Passage: {passage_2}
...
Title:
Passage: {passage_n}

Question: {query}
Answer:
```

My own Prompt

```
<|begin_of_text|><|user|>
DOCUMENT:{document}
QUESTION:{query}
INSTRUCTIONS:
Answer the user's QUESTION using the
DOCUMENT textabove.
Keep your answer grounded in the facts of
the DOCUMENT.
If the DOCUMENT doesn't contain thefacts to
answer the QUESTION, return{{NONE}}.
<|end_of_text|><|assistant|>
```

CoT Prompt

```
Title:
Passage: {passage_1}
Title:
Passage: {passage_2}
...
Title:
Passage: {passage_n}

Question: {query}
Let's think step by step.
Answer:
```

Metric	Original	Custom	CoT
Accuracy	0.02425	0.00107	0.01120
ROUGE-1	0.03126	0.00775	0.02148
ROUGE-2	0.00690	0.00027	0.00227
ROUGE-L	0.03111	0.00769	0.02109
ROUGE-Lsum	0.03127	0.00770	0.02119

- Custom Prompt :

- Lower evaluation scores
- Output often contains {{NONE}}

- CoT Prompt :

- Unexpectedly lower performance than default
- Model starts with "Step 1", but no "Step 2" or further reasoning

-〉Both prompt structure and model capacity influence the quality and depth of reasoning.

Results

2-1. LoRA finetuning in summary task

Metric	A	B
ROUGE-1	0.02061	0.00245
ROUGE-2	0.00001	0.0
ROUGE-L	0.01815	0.00245
ROUGE-Lsum	0.01976	0.00245

Table 4: Performance comparison between summary gpt model with LoRA finetuning(A) and without(B).

- ROUGE scores significantly improved after LoRA fine-tuning

From original summary model

```
Singing the national anthem is a risky
proposition . Whitney Houston nailed it;
Roseanne Barr destroyed it .
(omission until end)
and and and and and and...
```

From finetuned summary model

```
Singing the national anthem is a risky
proposition . Whitney Houston nailed it;
Roseanne Barr destroyed it .
(end of sentence)
```

Example 2: Summary model Output

- Reduced repetitive or unnatural endings (e.g., "and and and...")

→ LoRA fine-tuning effectively enhances GPT's generation quality!

Results

2-2. LoRA in RAG-finetuned model

Metric	A	B
Accuracy	0.00015	0.00015
ROUGE-1	0.00911	0.00064
ROUGE-2	0.0	0.0
ROUGE-L	0.00905	0.00065
ROUGE-Lsum	0.00909	0.00065

Table 5: Performance comparison between RAG - gpt model with LoRA finetuning(A) and without(B).

- LoRA fine-tuning significantly improved ROUGE scores while maintaining accuracy.

→ LoRA fine-tuning also effectively enhances RAG's generation quality as well!

Results

2-3. RAG-Finetuning after LoRA finetuning

Metric	A	B
Accuracy	0.0	0.00015
ROUGE-1	0.01109	0.00064
ROUGE-2	0.0	0.0
ROUGE-L	0.01107	0.00065
ROUGE-Lsum	0.01111	0.00065

Table 6: Performance comparison between RAG - gpt model after LoRA finetuning(A) and without(B).

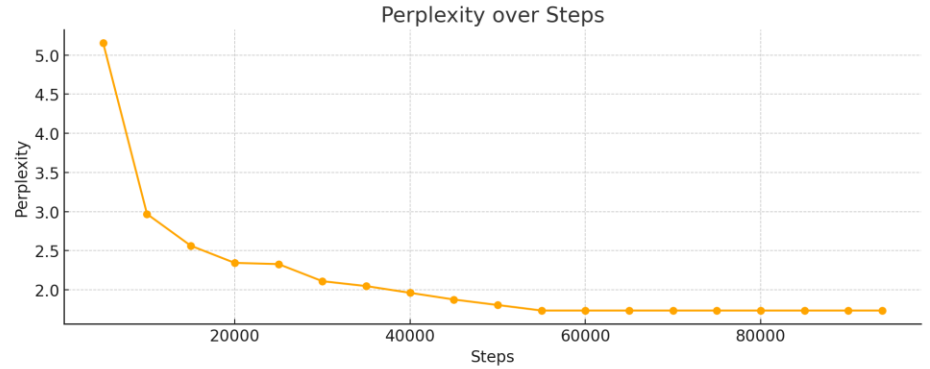
- When LoRA finetuning is done first, ROGUE score improved as well, but accuracy dropped.

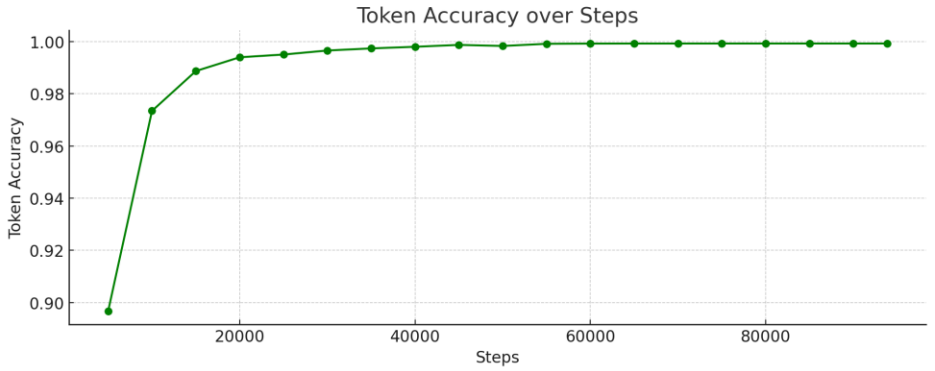
Limitation

1. Overfitted GPT model

Metric	Best Score
Loss	0.546875
Perplexity	1.734375
Token Accuracy	0.99927

Table 7: Best Score during pretraining





- Since GPT model has small amount of parameters and relatively small dataset, Overfitting is done during pretraining.

Limitation

2. Accuracy Degradation in the LoRA-RAG Model

Problem : After LoRA fine-tuning, a mismatch in embedding vocabulary size was observed.

Before finetuning : 50266

After finetuning : 50258

Although this was addressed by resizing the embedding layer, the slight discrepancy may have contributed to reduced accuracy.

(Ex. observed in Experiment 2-3)

Additionally, extremely low accuracy may also result from overfitting during pretraining.

Further experiments are required to identify the exact cause of the performance degradation.

Reflections

2. Accuracy Degradation in the LoRA-RAG Model

Problem : After LoRA fine-tuning, a mismatch in embedding vocabulary size was observed.

Before finetuning : 50266

After finetuning : 50258

Although this was addressed by resizing the embedding layer, the slight discrepancy may have contributed to reduced accuracy.

(Ex. observed in Experiment 2-3)

Additionally, extremely low accuracy may also result from overfitting during pretraining.

Further experiments are required to identify the exact cause of the performance degradation.

Thank you!

Default Project RAG / Improvement via LoRA Fine-Tuning