

DATA ENGINEERING MATTERS

Yu Kangqi 121090735

School of Management and Economics

Shenzhen, China

kangqiyu@link.cuhk.edu.cn

ABSTRACT

I design four data engineering methods, including none, to create datasets. These datasets are used to fine-tune a large language model. Evaluation shows that the latter two data engineering methods are effective.

1 INTRODUCTION

What is your research topic Data engineering plays a crucial role in fine-tuning large language models (LLMs). In this report, I delve into the significance of data engineering in fine-tuning LLMs and why it is essential for achieving better model performance.

Why the task is important Fine-tuning LLMs aims to transform them into domain-specific experts capable of understanding and generating contextually relevant text in a particular field. The knowledge these LLMs acquire is directly derived from the datasets they are trained on. Therefore, the quality and relevance of the training data play a crucial role in the success of fine-tuning efforts.

A well-organized and informative dataset is essential for fine-tuning models effectively. It provides necessary contexts, vocabularies, and domain-specific knowledge that LLMs need to excel in their designated area. Moreover, an accurate, diverse, and representative dataset ensures that the fine-tuned model can generalize well to real-world scenarios and produce reliable and contextually appropriate outputs.

In summary, the significance of data engineering in fine-tuning large language models lies in its ability to provide the foundation for model adaptation and optimization. By preparing high-quality and domain-specific datasets, researchers and practitioners can empower LLMs to become experts in their respective fields, unlocking their full potential for a wide range of natural language understanding and generation tasks.

What you did and what you achieved I design four types of data engineering methods (including no data engineering) to create four datasets. Then, I use these

datasets to fine-tune a large language model. Finally, I use two methods to evaluate these models, which indicate the latter two data engineering methods work.

2 EXPERIMENT DESIGN

Definition of the task To delve into the importance of data engineering, I use a Chinese financial dataset to work as the baseline. Then, I modify this dataset into several different formats. Comparing the models fine-tuned from them will determine whether data engineering can promote the fine-tuned model a lot.

How do you design the entire experiment? For this experiment, I use three ways to modify the original dataset:

- Add the first five questions and answers to all the data prompts as prerequisite knowledge.
- Add the five most common types of questions and answers to all the data prompts as prerequisite knowledge.
- Convert all these multiple-choice questions to blank-filled questions.

The first two modifications are to identify whether the prerequisite knowledge will help the model learn the intrinsic logic. The last one modification is to identify whether the wrong choices in the question will mislead the model.

The data items used for training, evaluation, and testing are intrinsically the same (processed from the same data) to ensure fairness.

3 CODE IMPLEMENTATION

3.1 EXPERIMENT SETTING

- Original dataset: FinGPT-fineval ¹
- Data engineering method: The outline is shown above, and the detail will be shown in the flowing subsections.
- Model: FreedomIntelligence/phoenix-inst-chat-7b²
- Fine-tune method: LoRA (Low-Rank Adaptation of Large Language Models) Hu et al. (2021) supported by PEFT (Parameter-Efficient Fine-Tuning) ³. The target modules of the model are query, key, and value.

¹<https://huggingface.co/datasets/FinGPT/fingpt-fineval>

²<https://huggingface.co/FreedomIntelligence/phoenix-inst-chat-7b>

³<https://huggingface.co/docs/peft/index>

- Test metric and test data: I have two metrics to evaluate the performance of these fine-tuned models. The first one is using the test questions of FinGPT-fineval (totally 265 questions) Hendrycks et al. (2021). The other one is using ChatGPT as an evaluator Wang et al. (2023). For the later one, the test data is provided by the course team of DDA6307 / CSC6052 / MDS6002 with 20 questions related to finance.

3.2 DATASET MODIFICATION

The below codes used some tools. For the detail of these tools, refer the file *data_tools.py* in the code attachment.

3.2.1 THE FIRST MODIFICATION

```
1 # load data from FinGPT
2 dataset = load_dataset("FinGPT/fingpt-fineval")
3 # store the pre knowledge before answer the question
4 pre_knowledgements = dataset["train"][:5]
5
6 #modify the dataset
7 modified_dataset = dataset.map(lambda batch:
    concatenate_pre_knowledgement(batch, pre_knowledgements), batched=
    True)
8 modified_train_data = modified_dataset['train']
```

3.2.2 THE SECOND MODIFICATION

```
1 # load data from FinGPT
2 dataset = load_dataset("FinGPT/fingpt-fineval")
3 # extract a label of which the exam is
4 labels = bind_labels(dataset["train"])
5 labels_counter = Counter(labels)
6 # select the most common labels out
7 most_common_labels = [t[0] for t in labels_counter.most_common(5)]
8 indices = [labels.index(label) for label in most_common_labels]
9 pre_knowledgements = dataset["train"][indices]
10
11 # modify the dataset
12 modified_dataset = dataset.map(lambda batch:
    concatenate_pre_knowledgement(batch, pre_knowledgements), batched=
    True)
13 modified_train_data = modified_dataset['train']
```

3.2.3 THE THIRD MODIFICATION

```

1  # load data from FinGPT
2  dataset = load_dataset("FinGPT/fingpt-fineval")
3  # get the correct choice
4  ans_list = []
5  for i in range(len(dataset['train']['output'])):
6  ans_list.append(match_choice(dataset['train']['output'][i]))
7  input_data = []
8  output_data = []
9  for i in range(len(dataset['train']['input'])):
10 input_data.append(dataset['train']['input'][i].split('\n')[0])
11 output_data.append(dataset['train']['input'][i].split('\n')[1 + ord(
    ans_list[i]) - ord("A")])
12 instruction = []
13 for i in range(len(dataset['train']['instruction'])):
14 instruction.append(dataset['train']['instruction'][i].split('单项')[0] +
    "填空题，请在_____上填空")

```

3.3 MODEL LOADING

```

1  # Some basic setting
2  model_id = "FreedomIntelligence/phoenix-inst-chat-7b"
3  rank = 8
4
5  # Quantization type (fp4 or nf4), According to QLoRA paper, for training
    4-bit base models (e.g. using LoRA adapters) one should use
6  bnb_4bit_quant_type = "nf4"
7
8  # Activate nested quantization for 4-bit base models (double quantization
    )
9  use_nested_quant = True
10
11 bnb_config = BitsAndBytesConfig(
12     load_in_4bit=True,
13     bnb_4bit_use_double_quant=use_nested_quant,
14     bnb_4bit_quant_type=bnb_4bit_quant_type,
15     bnb_4bit_compute_dtype=torch.bfloat16
16 )
17
18 tokenizer = AutoTokenizer.from_pretrained(model_id)
19 model = AutoModelForCausalLM.from_pretrained(model_id,
    quantization_config=bnb_config, device_map=device_map) # device_map
    ={"": 0}
20 model.gradient_checkpointing_enable()
21 model = prepare_model_for_kbit_training(model)
22

```

```
23 # You can try differnt parameter-effient strategy for model trianing, for  
    more info, please check https://github.com/huggingface/peft  
24 config = LoraConfig(  
25     r=rank,  
26     # empirical value for lora_alpha: 2*r  
27     lora_alpha=2*rank,  
28     target_modules=target_modules, # target_modules=["query_key_value"]  
29     lora_dropout=0.05,  
30     bias="none",  
31     task_type="CAUSAL_LM"  
32 )  
33  
34 model = get_peft_model(model, config)
```

3.4 MODEL TRAINING

The training parameters are not shown below. Because the capacity of the GPU is limited, I choose different parameters for different datasets to fine-tune. For details, you can refer the code attachment.

```
1 training_arguments = transformers.TrainingArguments(**train_args)  
2  
3 model.train()  
4 trainer = transformers.Trainer(  
5     model=model,  
6     tokenizer=tokenizer,  
7     args=training_arguments,  
8     train_dataset=train_dataset,  
9     eval_dataset=val_dataset,  
10    data_collator=data_collator  
11 )  
12  
13 trainer.train()
```

3.5 EVALUATION

3.5.1 USING MULTIPLE CHOICE QUESTIONS AS EVALUATORS

Just as I stated before, I use the test data part of FinGPT-fineval as exam question. Then, I used these models and GPT to answer the questions. Finally, I use the correctness as the performance evaluation metric.

3.5.2 USING GPT AS EVALUATOR

We use ChatGPT for evaluation with the following prompt:

We would like to request your feedback on the two AI assistants in response to the user question displayed above.

Please evaluate the helpfulness, relevance, accuracy, level of details of their responses. You should tell me whether Assistant 1 is ‘better than’, ‘worse than’, or ‘equal to’ Assistant 2.

Please first compare their responses and analyze which one is more in line with the given requirements.

In the last line, please output a single line containing only a single label selecting from ‘Assistant 1 is better than Assistant 2’, ‘Assistant 1 is worse than Assistant 2’, and ‘Assistant 1 is equal to Assistant 2’, avoiding any potential bias and ensuring that the order in which the responses were presented does not affect your judgment.

3.5.3 EVALUATION RESULTS

Following is the resultw of our fine-tuned models:

ChatGPT	winning rate	exam correctness
Ours (<i>original dataset fine-tuned</i>)	0.10	0.28
Ours (<i>modified dataset1</i>)	0.15	0.29
Ours (<i>modified dataset1</i>)	0.20	0.29
Ours (<i>modified dataset1</i>)	0.20	0.30
GPT (<i>gpt-3.5-turbo</i>)	/	0.50

表 1: Experimental Results

4 RESULT ANALYSIS

4.1 QUANTITATIVE EVALUATIONS

Our data engineering method works based on the above result. Both the winning rate compared with answers from GPT and their exam correctness are ascending. It indicates the data engineering method that adds prerequisite knowledge (more commonly used and more powerful) and drops the wrong information in the choices can improve the fine-tuning power. However, the best model is still worse than GPT (The model we trained is 7B, which is relatively much smaller than get-3.5-turbo).

4.2 CASE STUDY

Take the first question in the 20 questions as an example 1.

For the model fine-tuned by original dataset, we can see the response of it contained ”金融创新: 金融市场不断创新, 推动金融产品和服务的发展。”. As a student who studied

An example of fine-tuned models reacting to a question**Prompt:**

请将您对下面这个问题的思考完整且详细的回答出来。

Question:

金融市场的主要特点是什么？

Output from the model fine-tuned with original data:

金融市场是买卖金融资产的场所，其主要特点包括：1. 多样性：金融市场具有多样性，包括股票市场、债券市场、外汇市场、衍生品市场等。2. 开放性：金融市场是开放的，投资者可以随时进入或退出市场。3. 竞争性：金融市场具有竞争性，市场上的参与者众多，包括金融机构、投资者等。4. 风险性：金融市场存在风险，投资者需要谨慎评估风险并做出决策。5. 透明度：金融市场具有透明度，市场参与者需要公开披露信息，以便投资者做出决策。6. 全球化：金融市场具有全球化特征，投资者可以随时参与全球金融市场。7. 金融创新：金融市场不断创新，推动金融产品和服务的发展。

Output from the model fine-tuned with modified dataset1:

金融市场是买卖金融资产的场所，其主要特点包括：1. 多样性：金融市场具有多样性，包括股票市场、债券市场、外汇市场、衍生品市场等。2. 开放性：金融市场是开放的，投资者可以随时进入或退出市场。3. 竞争性：金融市场具有竞争性，市场参与者众多，包括金融机构、投资者等。4. 风险性：金融市场存在风险，投资者需要谨慎评估风险并做出决策。5. 透明度：金融市场具有透明度，市场参与者需要公开披露信息，以便投资者做出决策。6. 全球化：金融市场具有全球化特征，投资者可以随时参与全球金融市场。7. 金融创新：金融市场不断创新，推动金融产品和服务的发展。

Output from the model fine-tuned with modified dataset2:

金融市场是买卖金融资产的场所，其主要特点包括：1. 多样性：金融市场具有多样性，包括股票市场、债券市场、外汇市场、衍生品市场等。2. 开放性：金融市场是开放的，投资者可以随时进入或退出市场。3. 竞争性：金融市场具有竞争性，市场参与者众多，包括金融机构、投资者等。4. 风险性：金融市场存在风险，投资者需要谨慎评估风险并做出决策。5. 透明度：金融市场具有透明度，市场参与者需要公开披露信息，以便投资者做出决策。6. 全球化：金融市场具有全球化特征，投资者可以随时参与全球金融市场。

Output from the model fine-tuned with modified dataset3:

金融市场是买卖金融资产的场所，其主要特点包括：1. 多样性：金融市场具有多样性，包括股票市场、债券市场、外汇市场、衍生品市场等。2. 开放性：金融市场是开放的，投资者可以随时进入或退出市场。3. 竞争性：金融市场具有竞争性，市场参与者众多，包括金融机构、投资者等。4. 风险性：金融市场存在风险，投资者需要谨慎评估风险并做出决策。5. 透明度：金融市场具有透明度，市场参与者需要公开披露信息，以便投资者做出决策。6. 杠杆效应：金融市场具有杠杆效应，投资者可以利用杠杆放大投资收益或降低投资风险。7. 全球化：金融市场具有全球化特征，投资者可以随时参与全球金融市场交易。

图 1: An example prompt and responses from models

basic financial knowledge, I think these words are too tedious and fail to express their meaning. The model fine-tuned by modified dataset1 is almost the same as the original dataset, but it is more concise. Then, the model fine-tuned by the modified dataset2 drops the last point, I think this makes the response better. Finally, the model fine-tuned by the modified dataset3 adds a new point above leverage, which indicates the changing of data format helps it learn something new.

5 CONCLUSION

Data engineering really matters. In this report, I compare the performances of different data engineering methods (including no data engineering). Adding some commonly used information and dropping the wrong choices in the multiple choice question will help this dataset fine-tune a better model.

ACKNOWLEDGMENT

This is the Assignment3 for DDA6307 / CSC6052 / MDS6002, see details in <https://nlp-course-cuhksz.github.io/>.

REFERENCES

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. In *International Conference on Learning Representations*, 2021. URL <https://openreview.net/forum?id=d7KBjmI3GmQ>.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yanzhi Li, Shean Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *CoRR*, abs/2106.09685, 2021. URL <https://arxiv.org/abs/2106.09685>.

Jiaan Wang, Yunlong Liang, Fandong Meng, Zengkui Sun, Haoxiang Shi, Zhixu Li, Jinan Xu, Jianfeng Qu, and Jie Zhou. Is chatgpt a good nlg evaluator? a preliminary study, 2023.