

LLM Supervised Fine-tuning Report

周秉霖 (521030910361) 彭泉泉 (521020910182)

Deep Learning Course Project

Abstract

大型语言模型 (LLMs) 已经展示了在各种自然语言任务上出色的表现, 从而减少了对广泛特征工程的需求。一系列中英文开源语言模型 (ChatGPT, Llama, Internlm, Qwen 等) 崭露头角, 这些大型语言模型的训练过程通常分为两个主要阶段: 预训练 (Pre-train) 和微调 (Fine-tuning)。该大作业旨在研究两种常见大模型 Llama-7B[4] 和 Baichuan2-7B[5] 通过 SFT (Supervised Fine-tuning) 的方式在 UltraChat 数据集子集训练过后的评分表现, 同时也提升了两个大模型在涉及客观知识领域的多轮对话能力。本项目代码已开源:<https://github.com/Zhou-bl/LLM-SFT>

1 Method

1.1 Data Process

实验中选用的数据集是清华大学建立的 UltraChat[2]: UltraChat 是一个大规模的对话数据集, 包含了数百万个对话和数十亿个对话轮次。UltraChat 的特点在于它使用了多个 ChatGPT API 进行相互对话, 生成了大量的高质量对话数据。

出于训练资源和时间的限制, 我们最终选择 UltraChat Dataset 的子集 (主要来自于 Section I) 作为我们的 Training Set。如 Fig. 1 所示, Sector I 涉及关于世界的问题 (Questions about the World): 这部分对话来自于对现实世界中的概念、实体和对象相关的广泛询问。所涉及的主题涵盖科技、艺术、金融等多个领域。具体数据格式如 Fig. 2 所示:

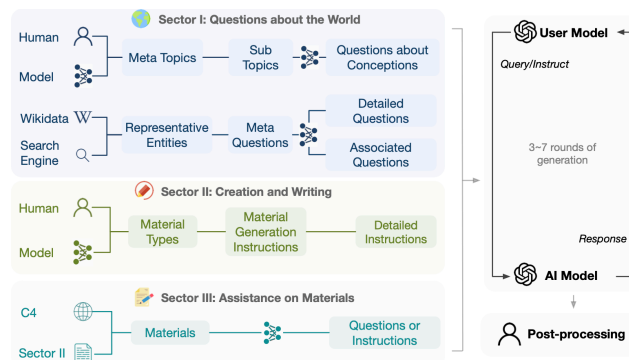


图 1: UltraChat 的构建过程中, 数据的三个 Sector 是从不同的 meta information 中得来。

```

{
  "id": "0",
  "data": [
    "How can cross training benefit groups like runners, swimmers, or weightlifters?",
    "Cross training can benefit groups like runners, swimmers, or weightlifters in the following ways:",
    "That makes sense. I've been wanting to improve my running time, but I never thought about incorpor",
    "Sure, here are some strength training exercises that can benefit runners: ...",
    "Hmm, I'm not really a fan of weightlifting though. Can I incorporate other forms of exercise into",
    "Yes, absolutely! ...",
    "..."
  ]
}

```

图 2: 每个 data sample 都是形如图中若干 sentence 构成的 list。它代表着大模型扮演的 Assitant 角色和 User 之间的一个长对话内容。

我们处理数据的步骤分为三个阶段：

1. 将过长的数据进行切分，只保留在给定长度范围内的数据作为训练数据。
2. 将数据 reformat 成对话形式。
3. 选取合适的 mask 方式，构造训练数据的 label。

接下来我们将重点介绍 2&3 阶段我们的处理方式。

常见的对多轮对话进行处理的方式有以下两种，但他们都有相应的缺点：

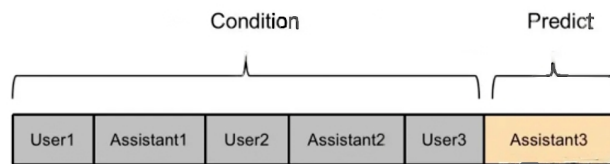


图 3: type1: 多轮对话只对 Assistant 最后的输出进行 predict，并计算 loss。缺点是无法对数据进行充分利用

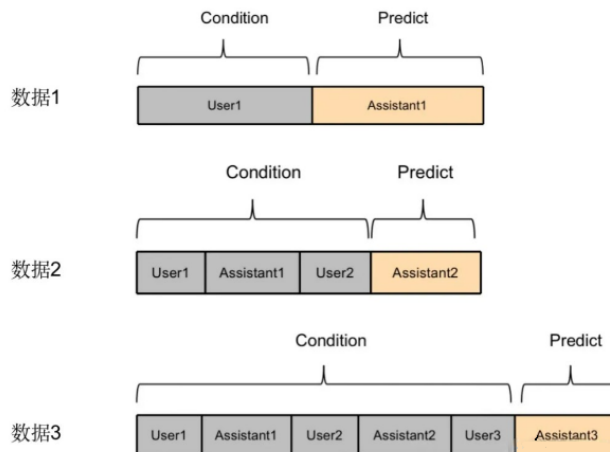


图 4: type2: 将每个多轮对话的数据点拆分成多个类似于 type1 的数据点。缺点是会让数据量扩大对话轮数倍

我们采用的是 Firefly[6] 提出的构建多轮对话的方式，将数据 reformat 成如 Fig. 5所示的形式。

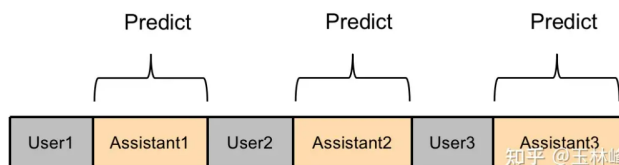


图 5: type3: 根据每个样本中 sentence 的来源来进行分类，只让模型预测来自于 Assistant 的句子，并且只有这些句子会参与 loss 的计算

这种方式既能让训练数据得到充分利用，又不会使训练数据膨胀。同时，因为我们使用的是因果语言模型，其 Attention mask 是一个对角掩码矩阵，使得每个 token 在编码时只能看到前面的 token。即：模型在预测 Assistant1 时，只能看到 User1 的内容，在预测 Assistant2 时只能看到 User1, Assistant1 和 User2 的内容 [7]。从而保证我们训练方法的正确性。

1.2 Train

我们使用 deepspeed[1] 框架部署模型并加速模型训练过程。具体的超参设置如下：

- Model Load Type: bfloat16
- Batch Size: 2
- Epoch: 3
- Max Length: 1024
- Optimizer: AdamW
- Learning Rate: Initial learning rate 10^{-5} , learning rate scheduler: cosine

1.3 Evaluation

我们使用 AlpacaFarm benchmark[3] 来对我们训练后的模型进行评估。AlpacaFarm 通过调用一个公认表现更好模型（例如 GPT-4, Claude 等）来对被测试模型的在数据集上的表现进行评估。具体来说：对于每一个问题，我们会给 GPT 两个对应的输出，其中一个来自于我们的模型对于该问题的输出，另一个是数据集中给定的回复，然后让 GPT 判断更加倾向于哪一个回复。我们用 GPT 对于模型在测试集上的输出的偏好率（win rate）来评估模型的好坏。win rate 越高模型表现越好。

在评测中，我们用 “gpt-3.5-turbo-instruct” 来作为我们的 evaluator，prompt 的设置如 Fig. 6.

2 Experiment Results

UltraChat	Llama-7B	Baichuan2-7B
w/o FineTune	13.91	15.92
SFT 1 epoch	21.45	24.38
SFT 2 epoch	16.15	26.15
SFT 3 epoch	16.92	25.62

表 1: Llama-7b 和 Baichuan2-7b 在 AlpacaFarm 下 win-rate 的表现，从表中可以看出经过 SFT 之后两个模型的表现都有一定程度的提升。


```

1 Human: Select the output (a) or (b) that best matches the given instruction.
2 Choose your preferred output, which can be subjective. Your answer should ONLY contain: Output (a) or Output (b). Here's an example:
3
4 # Example:
5 ## Instruction:
6 Give a description of the following job: "ophthalmologist"
7
8 ## Output (a):
9 An ophthalmologist is a medical doctor who specializes in the diagnosis and treatment of eye diseases and conditions.
10
11 ## Output (b):
12 An ophthalmologist is a medical doctor who pokes and prods at your eyes while asking you to read letters from a chart.
13
14 ## Which is best, Output (a) or Output (b)?
15 Output (a)
16
17 Here the answer is Output (a) because it provides a comprehensive and accurate description of the job of an ophthalmologist.
18 In contrast, output (b) is more of a joke. Now is the real task, remember to only include Output (a) or Output (b) in your answer, not the explanation.
19
20 # Task:
21 Now is the real task, do not explain your answer, just say Output (a) or Output (b).
22
23 ## Instruction:
24 {instruction}
25
26 ## Output (a):
27 {output_1}
28
29 ## Output (b):
30 {output_2}
31
32 ## Which is best, Output (a) or Output (b)?
33
34 Assistant:

```

图 6: prompt: 每次调用 evaluator 时的 prompt

我们的实验结果可参看1, 与其他模型的比较可参看 Fig. 7, 由图可见, 我们 SFT 在 Alpaca-Farm Benchmark 下成效显著: Llama-7b 在经过 1 epoch 的训练后效果逼近 Baichuan-13B-Chat, Baichuan2-7b 在经过 2 epoch 的训练后其效果远超 Baichuan-13B-Chat。

Alpaca 7B 	2 6. 4 6 %	396
Pythia 12B OASST SFT 	2 5. 9 6 %	726
Falcon 7B Instruct 	2 3. 6 0 %	478
Baichuan-13B-Chat 	2 1. 8 0 %	1727
Davinci001 	1 5. 1 7 %	296

[Github](#)

图 7: AlpacaFarm Leaderboard 上部分模型的表现

3 Conclusions

Baichuan2-7B 的性能在全面超过 Llama, 表现出色。对于 Llama-7b, Fine-tuning 到第 2, 3 个 epoch 的时候, 它的性能不如 epoch 1, Baichuan2 也在第 3 个 epoch 中的性能出现了下降。初步猜测数据集大小相对小, 模型可能在训练数据上出现了 overfitting。后续可能的检验方法可以通过观察 loss function 在 test dataset 上的变化来判断是否出现了过拟合, 同时性能评测指标也可以通过一些更加符合 SFT 的 benchmark 来说明 (AlpacaFarm Benchmark 更加适合说明 LLM alignment 的效果)。

Acknowledgements

感谢杨超老师和董智辰助教对本项目的指导和帮助。

参考文献

- [1] Reza Yazdani Aminabadi, Samyam Rajbhandari, Minjia Zhang, Ammar Ahmad Awan, Cheng Li, Du Li, Elton Zheng, Jeff Rasley, Shaden Smith, Olatunji Ruwase, and Yuxiong He. Deep-speed inference: Enabling efficient inference of transformer models at unprecedented scale, 2022.
- [2] Ning Ding, Yulin Chen, Bokai Xu, Yujia Qin, Zhi Zheng, Shengding Hu, Zhiyuan Liu, Maosong Sun, and Bowen Zhou. Enhancing chat language models by scaling high-quality instructional conversations, 2023.
- [3] Kiwan Maeng, Alexei Colin, and Brandon Lucia. Alpaca: Intermittent execution without checkpoints. *Proceedings of the ACM on Programming Languages*, 1(OOPSLA):1–30, 2017.
- [4] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. Llama: Open and efficient foundation language models, 2023.
- [5] Aiyuan Yang, Bin Xiao, Bingning Wang, Borong Zhang, Ce Bian, Chao Yin, Chenxu Lv, Da Pan, Dian Wang, Dong Yan, Fan Yang, Fei Deng, Feng Wang, Feng Liu, Guangwei Ai, Guosheng Dong, Haizhou Zhao, Hang Xu, Haoze Sun, Hongda Zhang, Hui Liu, Jiaming Ji, Jian Xie, JunTao Dai, Kun Fang, Lei Su, Liang Song, Lifeng Liu, Liyun Ru, Luyao Ma, Mang Wang, Mickel Liu, MingAn Lin, Nuolan Nie, Peidong Guo, Ruiyang Sun, Tao Zhang, Tianpeng Li, Tianyu Li, Wei Cheng, Weipeng Chen, Xiangrong Zeng, Xiaochuan Wang, Xiaoxi Chen, Xin Men, Xin Yu, Xuehai Pan, Yanjun Shen, Yiding Wang, Yiyu Li, Youxin Jiang, Yuchen Gao, Yupeng Zhang, Zenan Zhou, and Zhiying Wu. Baichuan 2: Open large-scale language models, 2023.
- [6] Jianxin Yang. Firefly(流萤): 中文对话式大语言模型. <https://github.com/yangjianxin1/Firefly>, 2023.
- [7] Linfeng Yu. Chatglm2-6b 多轮对话训练方式. <https://zhuanlan.zhihu.com/p/651293366>, 2023.