Better supervised fine-tuning of closed-source large models

Fei Ding[1], **Correspondence:** dingfei@email.ncu.edu.cn

Abstract

The recent proliferation of so-called open-source large language models (such as LLaMA, Falcon, Mistral) has introduced a broader range of alternatives for AI practitioners and researchers. However, the majority of these models cannot be considered truly open-source, as they often provide only partial artifacts, such as final model weights or inference code. Furthermore, technical documentation accompanying these models tends to focus on high-level architectural decisions and superficial metrics, leaving critical aspects of the training process, including dataset composition, distribution, model checkpoints, and intermediate results, largely undisclosed. This lack of transparency presents a significant barrier to progress in the field, restricting the potential for open, collaborative research. In the absence of access to original datasets, attempts to further train or fine-tune these models by third parties are susceptible to issues such as catastrophic forgetting.In response to this challenge, we propose a method that facilitates more effective supervised fine-tuning of these closed-source models, without requiring access to the original data, while mitigating the risk of catastrophic forgetting.

1 Introduction

Catastrophic forgetting represents a critical challenge for large language models (LLMs) and neural networks (NNs). This phenomenon is characterized by the models' propensity to abruptly lose previously acquired knowledge when assimilating new information. Such a limitation significantly impedes the development of robust and reliable artificial intelligence systems, particularly in dynamic contexts where ongoing learning from novel data is imperative.

Catastrophic forgetting—the tendency of deep neural networks to "forget" previously acquired knowledge when introduced to new information—has been a subject of investigation since 1989 McCloskey and Cohen, 1989. This phenomenon is most evident when models are sequentially trained on distinct tasks; however, it also occurs whenever a model learns information in a sequential manner, particularly when there are shifts in data distribution over time. In practical machine learning applications, it is common for new training data to be introduced continuously. To incorporate this new information into model training, developers face a choice: they can either retrain the entire model from scratch, starting with randomly initialized weights and utilizing all available training data, a process that is computationally intensive, or they can take an existing model trained on prior data and perform fine-tuning on the newly acquired data. However, since new data typically originates from a distribution that is slightly different from that of the old data, significant changes in distribution can exacerbate the effects of catastrophic forgetting during the fine-tuning process.

The landscape of Large Language Models (LLMs) has undergone a remarkable transformation over the past year, characterized by an unprecedented surge in both their popularity and capabilities. Leading this evolution are proprietary LLMs such as GPT-4 OpenAI, 2023 and Claude Claude, 2023, which have garnered significant attention within the AI community owing to their exceptional power and versatility. Concurrently, the recent emergence of openly accessible yet highly capable LLMs, including LLaMA (Touvron et al., 2023a,b), Falcon (Penedo et al., 2023), and Mistral (Jiang et al., 2023), has empowered researchers and practitioners to easily acquire, customize, and deploy LLMs across a broader range of environments and applications.

Catastrophic forgetting and overtraining (or overfitting) represent distinct challenges encountered in the training of neural networks and large language models. Catastrophic forgetting occurs when a model discards previously acquired knowledge upon assimilating new information, particularly in sequential learning contexts. This phenomenon is attributed to the modifications in model weights that disrupt the performance of earlier tasks. In contrast, overtraining arises when a model becomes excessively attuned to the training data, leading it to capture noise and specific details rather than generalizable patterns, ultimately resulting in poor performance on new, unseen data. While catastrophic forgetting undermines knowledge retention in dynamic learning environments, overfitting significantly restricts the model's ability to generalize effectively from the training set to novel data.

When continuing training, in order to address both catastrophic forgetting and overfitting, it requires us to have knowledge of both the original data and its distribution.

Despite the increasing prominence and accessibility of open-source large language models (LLMs), a significant trend has emerged towards restricting visibility and access to the intricacies of their training, fine-tuning, and evaluation methodologies. This includes critical components such as the underlying training code and datasets, which are essential for a comprehensive understanding of model behavior and performance.

This approach limits our ability to perform SFT (Supervised Fine-Tuning) on these models.

Because using the same data easily leads to overfitting, while differences in data distribution can cause catastrophic forgetting, better SFT (Supervised Fine-Tuning) requires an alternative approach for models that do not disclose their original SFT data. We can reverse-engineer the model parameters to extract the distribution of the original SFT data, then generate new SFT data based on this distribution, and mix it with our own SFT data in a certain proportion. This allows for more effective fine-tuning.

This paper presents the following contributions:

- We deciphered the hidden data distribution of open-source models through model parameters and used it for experience replay during SFT fine-tuning to better mitigate catastrophic forgetting.

- We obtained the optimal instruction responses through mutual scoring among three models, significantly improving the response quality and enhancing the effectiveness of SFT.

2 Background and Related Work

2.1 Data Rehearsal

Robins (ROBINS, 1995) introduced the concept of rehearsal in 1995, shortly following the advent of the notion of catastrophic forgetting. In essence, this approach entails incorporating data from previous tasks during the training of new ones. While this method has demonstrated considerable efficacy, it necessitates maintaining access to historical data, or at the very least, an independent and identically distributed (i.i.d.) subsample of such data, which may not always be feasible. Furthermore, integrating past data increases the overall volume of training data, resulting in longer training durations for each epoch during model fine-tuning.

Since most large models do not have publicly available datasets for rehearsal, the common approach is to use some public sft datasets mixed with their own sft datasets to simulate a review process. However, this approach can lead to certain issues. Our approach involves extracting the concealed data distribution of the supervised fine-tuning (SFT) instructions directly from the model parameters.

2.2 Continue Fine-tuning

Our methodology addresses the challenge of continual fine-tuning, wherein the model undergoes successive fine-tuning with newly acquired data post-initial fine-tuning. Continual learning is essential for models that must adapt to dynamic environments, assimilating information from a continuous data stream while retaining previously learned knowledge. A critical obstacle in this domain is the issue of catastrophic forgetting, which refers to the pronounced degradation in performance on earlier tasks when the model is exposed to novel data. As the model adjusts its parameters to incorporate new information, it inadvertently overwrites previously acquired knowledge, thereby diminishing its effectiveness on prior tasks. To address this, the research community has proposed a range of strategies, typically classified into four main categories: Replay-Based (Shin et al., 2017; Ren et al., 2024), Regularization-Based (Mi et al., 2020), Gradient-Based (Lee et al., 2021), and Architecture-Based (Geng et al., 2021) approaches. In our experiments, we adopted a basic experience replay mechanism, reduced the initial learning rate to avoid overfitting.

3 Methods

Our experiments are divided into three parts: the first part involves extracting the original SFT data distribution from the model; the second part mixes the extracted SFT data with new data for training; and the third part uses commonly available general SFT data mixed with new data for training, comparing the results with those from the second part.

3.1 Extracting the instruction distribution.

Cracking the instruction distribution consists of three steps: (1) instruction generation, (2) response generation, and (3) filtering high-quality responses. The pipeline can be fully automated without any human intervention.

Step 1: Instruction Generation.

The objective of this step is to extract unreleased training data from the model's parameters. Given an open-weight aligned large language model (e.g., Llama-3-70B-Instruct), we design a pre-query template in the format of the predefined instruction template.

We input the prompt "<|start_header_id|>user<|end_header_id|>" into the large model (Llama-3-70B-Instruct), which generates a single instruction in response. By repeating this process 100,000 times, we obtain a total of 100,000 instructions, which collectively represent the current instruction distribution of the large model.

Step 2: Response Generation. The objective of this step is to generate responses to the instructions obtained in Step 1.

We send these instructions to Llama-3-70B-Instruct and two additional powerful large language models(such as gpt4 and Qwen2-72B-Instruct). For each instruction, each model generates three responses, resulting in a total of nine responses for each instruction.

Step 3: Filtering High-quality Responses. For each instruction, we evaluate nine generated responses using the three models previously mentioned, assigning quality scores to each response. The scores from the three models are then averaged to identify the response with the highest overall score.

Combining the optimal response with the corresponding instruction forms the instruction dataset. The exact prompt we use for scoring is provided in Table 2.

3.2 Data mixing and training.

Mix the extracted SFT data with our new SFT data, then proceed with training. The new SFT data accounts for 17% of all the data. The learning rate is set to 1e-6.

3.3 Comparative experiment.

Use other open-source SFT datasets instead of the extracted SFT data for comparative experiments to identify which dataset used for experience replay results in less catastrophic forgetting.

Baselines for Supervised Fine-Tuning and Preference Optimization. These datasets include: **Evol Instruct** (Xu et al., 2023), **UltraChat** (Ding et al., 2023), **ShareGPT** (Chiang et al., 2023), **WildChat** (Zhao et al., 2024),**GenQA** (Chen et al., 2024), **OpenHermes 1** (Teknium, 2023b), **OpenHermes 2.5** (Teknium, 2023a), and **Tulu V2 Mix** (Ivison et al., 2023). ShareGPT and WildChat are representative human-written datasets containing 112K and 652K high-quality multi-round conversations between humans and GPT, respectively. Evol Instruct, UltraChat, and GenQA are representative open-source synthetic datasets. Following (Meng et al., 2024), we use the 208K sanitized version of Ultrachat provided by HuggingFace[1]. OpenHermes 1, OpenHermes 2.5, and Tulu V2 Mix are crowd-sourced datasets consisting of a mix of diverse open-source instruction datasets, with 243K, 1M, and 326K conversations, respectively.

We evaluated a variety of tasks featured on the Hugging Face Open LLM Leaderboard (Beeching et al., 2023), as presented in Table 1. The tasks include MMLU-PRO (Massive Multitask Language Understanding - Professional) (Wang et al., 2024), GPQA (Graduate-Level Google-Proof Q&A Benchmark) (Rein et al., 2023), IFEval (Zhou et al., 2023) and MATH level 5 (Hendrycks et al., 2021). Our experimental results demonstrate that employing our approach (extracting instruction distributions from the model) yields improved fine-tuning performance.

3.4 Ablation Study

We tested the responses generated directly by the target model without using the three models for filtering, and the results are presented in Table 1.We also tested generating three responses solely by the target model without using the other two models

[1] https://huggingface.co/datasets/HuggingFaceH4/ultrachat_200k

Alignment Setup	MMLU-PRO (5)	GPQA (0)	IFEval(0)	Math Lvl 5 (4)	Average
Llama-3-70B-Instruct	46.74	4.92	80.99	23.34	**39.00**
Extracted-Instructions-Unfiltered	46.11	4.72	81.31	23.11	38.81
One-Model-Filtered	46.55	4.91	81.72	23.06	39.06
Three-Models-Mix-Filtered	46.73	4.88	81.93	23.29	**39.21**
ShareGPT	46.14	4.31	81.31	20.24	38.00
Evol Instruct	45.76	4.64	82.52	22.30	38.81
GenQA	43.33	4.48	80.43	15.41	35.91
OpenHermes 1	45.31	4.21	81.91	15.52	36.74
OpenHermes 2.5	45.63	4.79	82.33	15.62	37.09
Tulu V2 Mix	46.47	4.19	82.69	16.62	37.49
WildChat	45.83	4.12	81.32	22.11	38.35
UltraChat	45.15	4.08	81.57	20.31	37.78

Table 1: This table compares the performance of models fine-tuned with supervision using the extracted instruction dataset for experience replay against baseline models and the official instruction model across various downstream benchmarks. All models are fine-tuned with supervision on the Llama-3-70B-Instruct model.

and then selecting the best one,and the results are presented in Table 1.

4 Conclusion

In this paper, we developed a method to extract instruction distributions from a model trained on an unpublished instruction dataset. We then leveraged two additional powerful models to collaboratively generate high-quality responses, forming an instruction dataset used as experience replay data during model fine-tuning. Compared to other baseline methods, our approach mitigates catastrophic forgetting and enhances fine-tuning performance.

5 Limitations

We conducted experiments only on Llama-3-70B-Instruct, achieving favorable results. Due to computational constraints, we did not perform extensive testing on other models.

References

Edward Beeching, Clémentine Fourrier, Nathan Habib, Sheon Han, Nathan Lambert, Nazneen Rajani, Omar Sanseviero, Lewis Tunstall, and Thomas Wolf. 2023. Open llm leaderboard.

Jiuhai Chen, Rifaa Qadri, Yuxin Wen, Neel Jain, John Kirchenbauer, Tianyi Zhou, and Tom Goldstein. 2024. Genqa: Generating millions of instructions from a handful of prompts. *arXiv preprint arXiv:2406.10323*.

Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E. Gonzalez, Ion Stoica, and Eric P. Xing. 2023. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality.

Claude. 2023. Claude 2.1 model card. Technical report, Claude Inc.

Ning Ding, Yulin Chen, Bokai Xu, Yujia Qin, Zhi Zheng, Shengding Hu, Zhiyuan Liu, Maosong Sun, and Bowen Zhou. 2023. Enhancing chat language models by scaling high-quality instructional conversations. *arXiv preprint arXiv:2305.14233*.

Binzong Geng, Fajie Yuan, Qiancheng Xu, Ying Shen, Ruifeng Xu, and Min Yang. 2021. Continual learning for task-oriented dialogue system with iterative network pruning, expanding and masking. *arXiv preprint arXiv:2107.08173*.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. 2021. Measuring mathematical problem solving with the math dataset. *arXiv preprint arXiv:2103.03874*.

Hamish Ivison, Yizhong Wang, Valentina Pyatkin, Nathan Lambert, Matthew Peters, Pradeep Dasigi, Joel Jang, David Wadden, Noah A Smith, Iz Beltagy, et al. 2023. Camels in a changing climate: Enhancing lm adaptation with tulu 2. *arXiv preprint arXiv:2311.10702*.

Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. 2023. Mistral 7b. *arXiv preprint arXiv:2310.06825*.

Seanie Lee, Hae Beom Lee, Juho Lee, and Sung Ju Hwang. 2021. Sequential reptile: Inter-task gradient alignment for multilingual learning. *arXiv preprint arXiv:2110.02600*.

Michael McCloskey and Neal J. Cohen. 1989. Catastrophic interference in connectionist networks: The sequential learning problem. 24:109–165.

Yu Meng, Mengzhou Xia, and Danqi Chen. 2024. Simpo: Simple preference optimization with a reference-free reward. *arXiv preprint arXiv:2405.14734*.

Fei Mi, Liangwei Chen, Mengjie Zhao, Minlie Huang, and Boi Faltings. 2020. Continual learning for natural language generation in task-oriented dialog systems. *arXiv preprint arXiv:2010.00910*.

OpenAI. 2023. Gpt-4 technical report.

Guilherme Penedo, Quentin Malartic, Daniel Hesslow, Ruxandra Cojocaru, Alessandro Cappelli, Hamza Alobeidli, Baptiste Pannier, Ebtesam Almazrouei, and Julien Launay. 2023. The refinedweb dataset for falcon llm: outperforming curated corpora with web data, and web data only. *arXiv preprint arXiv:2306.01116*.

David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R Bowman. 2023. Gpqa: A graduate-level google-proof q&a benchmark. *arXiv preprint arXiv:2311.12022*.

Weijieying Ren, Xinlong Li, Lei Wang, Tianxiang Zhao, and Wei Qin. 2024. Analyzing and reducing catastrophic forgetting in parameter efficient tuning. *arXiv preprint arXiv:2402.18865*.

ANTHONY ROBINS. 1995. Catastrophic forgetting, rehearsal and pseudorehearsal. *Connection Science*, 7(2):123–146.

Hanul Shin, Jung Kwon Lee, Jaehong Kim, and Jiwon Kim. 2017. Continual learning with deep generative replay. *Advances in neural information processing systems*, 30.

Teknium. 2023a. Openhermes 2.5: An open dataset of synthetic data for generalist llm assistants.

Teknium. 2023b. Openhermes dataset.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023a. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023b. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyan Jiang, et al. 2024. Mmlu-pro: A more robust and challenging multi-task language understanding benchmark. *arXiv preprint arXiv:2406.01574*.

Can Xu, Qingfeng Sun, Kai Zheng, Xiubo Geng, Pu Zhao, Jiazhan Feng, Chongyang Tao, and Daxin Jiang. 2023. Wizardlm: Empowering large language models to follow complex instructions. *arXiv preprint arXiv:2304.12244*.

Wenting Zhao, Xiang Ren, Jack Hessel, Claire Cardie, Yejin Choi, and Yuntian Deng. 2024. Wildchat: 1m chatGPT interaction logs in the wild. In *The Twelfth International Conference on Learning Representations*.

Jeffrey Zhou, Tianjian Lu, Swaroop Mishra, Siddhartha Brahma, Sujoy Basu, Yi Luan, Denny Zhou, and Le Hou. 2023. Instruction-following evaluation for large language models. *arXiv preprint arXiv:2311.07911*.

A Appendix

Below is a user instruction and an AI response. Evaluate the quality of the AI's response based on how well it fulfills the user's request. Assign a score based on the following 5-point scale: 1: The response is incomplete, off-topic, or contains irrelevant, vague, or missing information. It may repeat the user's question, include personal opinions, or be written from a non-AI perspective (e.g., blog-like). It may also have promotional or irrelevant content. 2: The response addresses some of the user's request but lacks detail or direct relevance. It provides only a general approach instead of a specific solution. 3: The response is helpful but lacks an AI perspective. It covers the user's request but appears taken from a personal blog, webpage, or similar source. It may include personal opinions, experiences, or mentions of external content. 4: The response is clear, complete, and written from an AI's perspective. It directly addresses the user's request, but there may be minor room for improvement, such as clarity or conciseness. 5: The response is excellent, written from an AI's perspective, with a clear focus on the user's request. It is thorough, well-organized, and shows expert knowledge without irrelevant content. The response is logical, easy to follow, and engaging. Provide a brief justification for your score and then write "Score: <rating>" in the last line. <generated instruction> <output>

Table 2: A prompt used to evaluate the quality of a response.