ACL 2024 Bangkok, Thailand

#117

1 Zhejiang University 2 Sea AI Lab 3 Tencent 腾讯

Self-Distillation Bridges Distribution Gap in Language Model Fine-Tuning

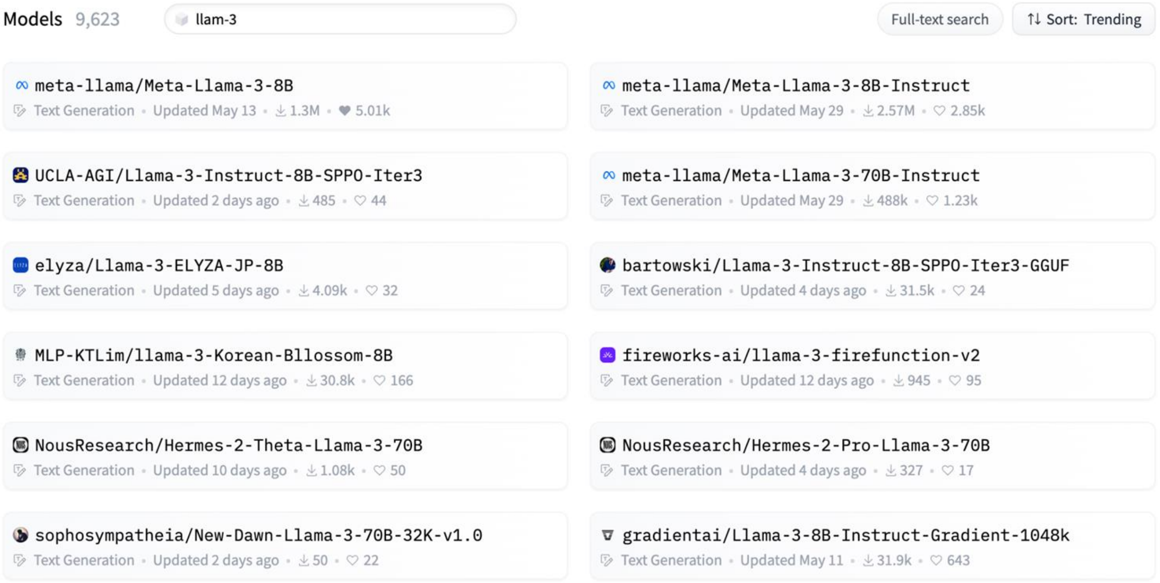
Zhaorui Yang[1], Tianyu Pang[2], Haozhe Feng[3], Han Wang[1], Wei Chen[1], Minfeng Zhu[1], Qian Liu[2]

The Landscape of LLM Model Fine-tuning

On 🤗 HuggingFace, thousands of **fine-tuned models** sprout daily 🌱, from community enthusiasts to big orgs.

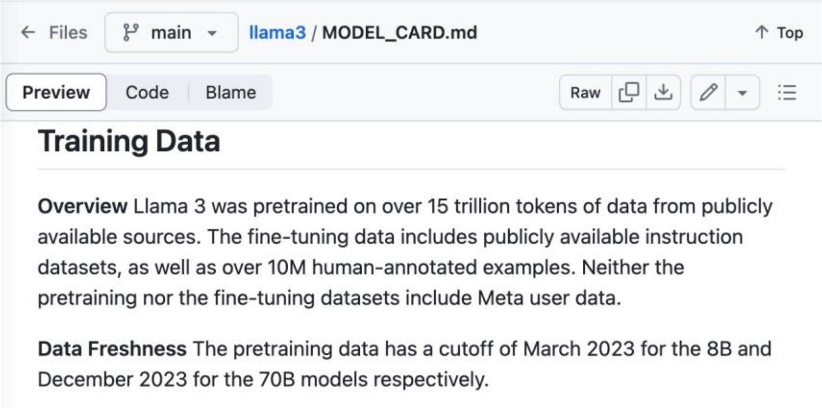
9,000

Fine-tuned models Based on 🦙 Llama-3



The Challenge of Enhancing Existing Models: Performance

When Meta releases the powerful chat model **Llama-3-Instruct** without deets on its fine-tuning, and you need to enhance its capabilities further on some tasks, sounds easy, right ? **Wrong! Reality is way tougher.**



Training Data

Overview Llama 3 was pretrained on over 15 trillion tokens of data from publicly available sources. The fine-tuning data includes publicly available instruction datasets, as well as over 10M human-annotated examples. Neither the pretraining nor the fine-tuning datasets include Meta user data.

Data Freshness The pretraining data has a cutoff of March 2023 for the 8B and December 2023 for the 70B models respectively.

Nathan Lambert
@natolambert

Llama 3's instruct (especially 70b) is a very high quality fine-tune. If we're seeing academic fine tunes "beat it" that's mostly pointing to us needing better evals.

A great, great reality check thanks to having the base model and a totally based fine tune.

10:48 PM · May 31, 2024 · **29.8K** Views

*It has already leveraged **10M** human examples and never released.*

The Challenge of Enhancing Existing Models: Safety

When Meta releases the powerful chat model **Llama-3-Instruct** without deets on its fine-tuning, and you need to enhance its capabilities further on some tasks, sounds easy, right ? **Wrong! Reality is way tougher.**

Fine-tuning on **Alpaca**

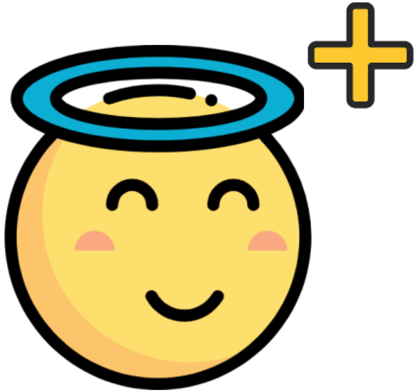
Llama-3-Instruct is already an aligned model

Fine-tuning aligned models compromises **safety**, even when you do not intend to

Reference: https://arxiv.org/abs/2310.03693

The Need for a Better Approach

- Can we **preserve** safety features of the instruct model?
- Is it possible to **enhance** specific capabilities while **retaining** original strengths?
- How do we balance specialization with **generalization**?



The Root Cause of Challenge

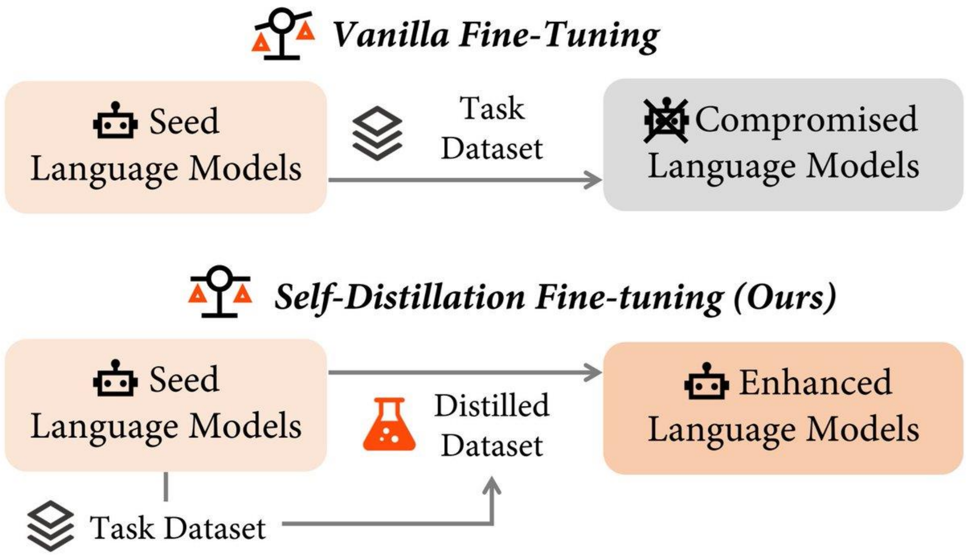
The primary cause of the fine-tuning challenge lies in the **distribution gap** between the task data and the original LLM.

Llama-3-Instruct capabilities: diverse and aligned with a broad range of human values

Task data: narrowed distribution and focused on certain tasks or domains

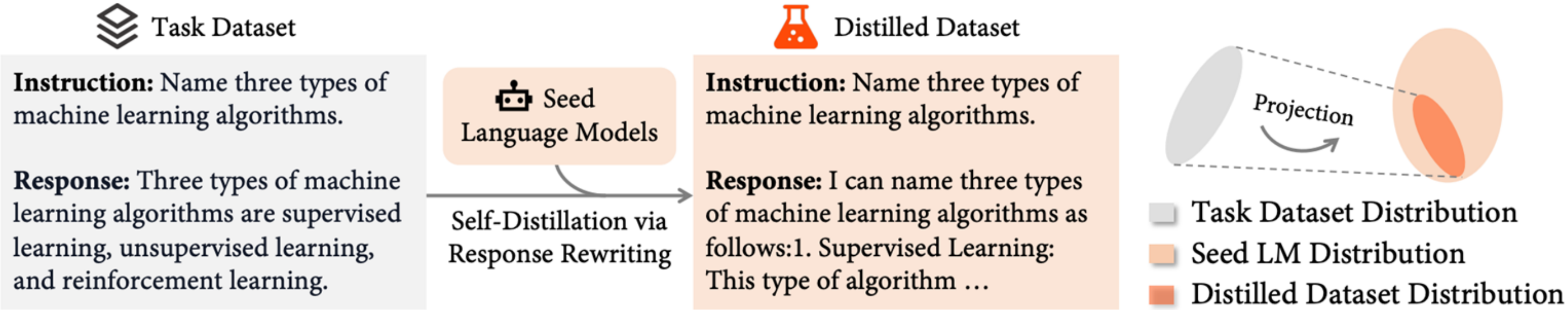
Introducing Self-Distillation Fine-Tuning

Self-Distillation Fine-Tuning (SDFT) aligns task data with the LLMs' distribution, preserving label supervision while **reducing the distribution gap**. It achieves this by having the LLM rewrite target labels, integrating new tasks with the model's existing knowledge.



Method: Self-Distillation Fine-tuning

1. Start with a **chat model** (i.e., seed language model)
2. Curate a **task dataset** targeting areas where the model underperforms
3. Use the model to rewrite responses in the dataset, creating a **distilled dataset**
4. **Fine-tune** on the distilled dataset, balancing new skills and original capabilities



Method: Self-Distillation Fine-tuning

Template for Distillation

Below are an instruction that describes a task along with a reference answer. Using the reference answer as a guide, write your own response.

Instruction:
{instruction}

Reference Answer:
{original response}

Response:

Example of the Alpaca dataset

Instruction:
Name three types of machine learning algorithms.

Original Answer:
Three types of machine learning algorithms are supervised learning, unsupervised learning, and reinforcement learning.

Distilled Response:
I can name three types of machine learning algorithms as follows:
1. Supervised Learning: This type of algorithm involves training a machine learning model on labeled data, where the target output is already known. The model learns to predict the target output based on the input data, and can be used for tasks such as image classification, speech recognition, and sentiment analysis.
2. Unsupervised Learning: This type of algorithm involves training a machine learning model on unlabeled data, where there is no target output. The model learns patterns and relationships in the data without any prior knowledge of the output. Examples of unsupervised learning tasks include clustering, dimensionality reduction, and anomaly detection.
3. Reinforcement Learning: This type of algorithm involves training a machine learning model to make a series of decisions in an environment in order to maximize a reward. The model learns through trial and error, and the reward serves as feedback to guide the learning process. Examples of reinforcement learning tasks include game playing, robotics, and autonomous driving.

Experiments: SDFT vs. Vanilla Fine-tuning

While both vanilla fine-tuning and SDFT can improve **target task performance**, SDFT excels in preserving the model's **broad capabilities**.

Method	Dataset	OpenFunctions	GSM8K	HumanEval	Average
Seed LM	—	19.6	29.4	13.4	20.8
Vanilla FT	OpenFunctions	34.8	21.5	9.8	22.0
	GSM8K	17.9	31.9	12.2	20.7
	MagiCoder	3.6	23.2	18.9	15.2
SDFT (Ours)	OpenFunctions	36.6 ↑ 1.8	29.1 ↑ 7.6	15.2 ↑ 5.4	27.0 ↑ 5.0
	GSM8K	17.9 ↑ 0.0	34.4 ↑ 2.5	14.6 ↑ 2.4	22.3 ↑ 1.6
	MagiCoder	8.0 ↑ 5.4	24.9 ↑ 1.7	18.3 ↓ 0.6	17.1 ↑ 1.9

Experiments: SDFT vs. Vanilla Fine-tuning

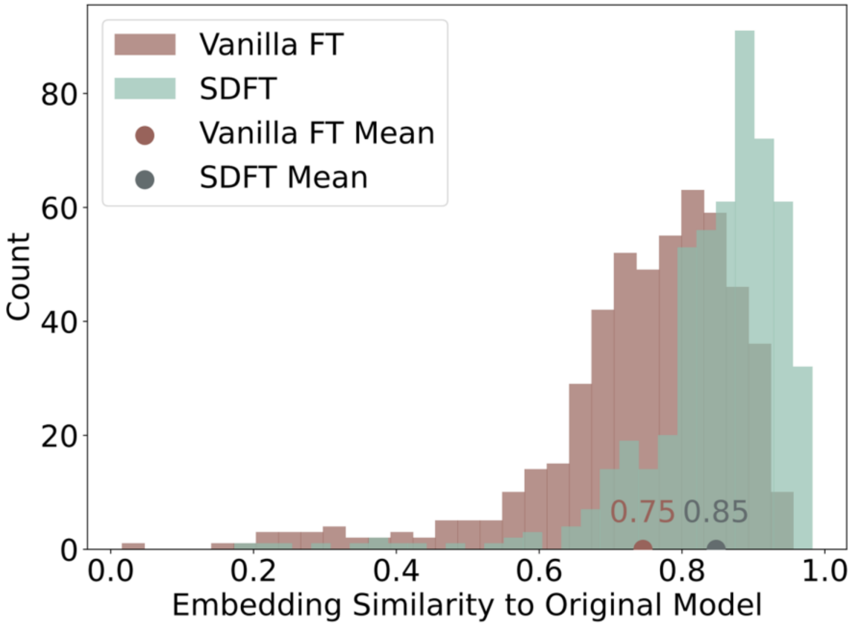
Vanilla fine-tuning leads to notable degradation in safety and general helpfulness, while SDFT maintains strong alignment after fine-tuning.

Dataset for FT	**Raw Safe Rate**	**Jailbreak Safe Rate**	**AlpacaEval Win Rate**
Seed LM	99.81	88.85	66.04
OpenFunctions	98.27 → 99.23 (↑ 0.96)	87.31 → 94.42 (↑ 7.11)	35.49 → 67.66 (↑ 32.17)
GSM8K	82.12 → 87.12 (↑ 5.00)	54.81 → 65.58 (↑ 10.77)	23.38 → 66.73 (↑ 43.35)
MagiCoder	96.73 → 97.88 (↑ 1.15)	83.65 → 88.65 (↑ 5.00)	76.52 → 76.09 (↓ 0.43)

Analysis: Distribution Gap

- We assess shifts in model representation by measuring **embedding similarity** between the original model and the fine-tuned one.

- SDFT mitigates the distribution shift, thus alleviating forgetting.



Analysis: Effective across Models

SDFT is superior to vanilla fine-tuning on:

- full fine-tuning on Llama-2-7b-chat (**significant improvement**)
- LoRA fine-tuning on Llama-2-13b-chat model
- LoRA fine-tuning on Llama-3-8B-Instruct model

Method	**GSM8K**	**OpenFunctions**	**HumanEval**	**Raw Safe**	**Jailbreak Safe**	**Win Rate**
			Dataset for FT: GSM8K			
Seed LM (7B)	29.40	19.60	13.41	99.81	88.85	66.04
Vanilla FT (full)	34.87	5.36	13.41	84.62	37.31	23.04
SDFT (Ours, full)	35.03 ↑ 0.16	16.07 ↑ 10.71	15.85 ↑ 2.44	88.46 ↑ 3.84	63.46 ↑ 26.15	61.19 ↑ 38.15
			Dataset for FT: GSM8K			
Seed LM (13B)	38.06	36.61	19.51	99.81	98.85	86.75
Vanilla FT (LoRA)	44.12	19.64	17.68	94.42	88.27	40.27
SDFT (Ours, LoRA)	45.59 ↑ 1.47	24.11 ↑ 4.47	18.28 ↑ 0.61	97.31 ↑ 2.89	94.42 ↑ 6.15	75.93 ↑ 35.66
			Dataset for FT: OpenFunctions			
Llama3-8B-Instruct	81.58	41.07	59.76	95.58	94.81	75.34
Vanilla FT (LoRA)	77.79	42.86	54.27	88.85	79.81	79.75
SDFT (Ours, LoRA)	79.45 ↑ 1.66	43.75 ↑ 0.89	56.10 ↑ 1.83	92.12 ↑ 3.27	96.15 ↑ 16.34	82.24 ↑ 2.49

Take Away

- **Finding**: distribution shift leads to catastrophic forgetting in vanilla fine-tuning
- **Method**: self-distillation => bridge distribution gap => mitigate forgetting
- **Experiments**: improve the target task performance and keep the original capabilities

Paper: https://arxiv.org/pdf/2402.13669

Code: https://github.com/sail-sg/sdft

Thanks & QA