Training and deploying open-source LLMs

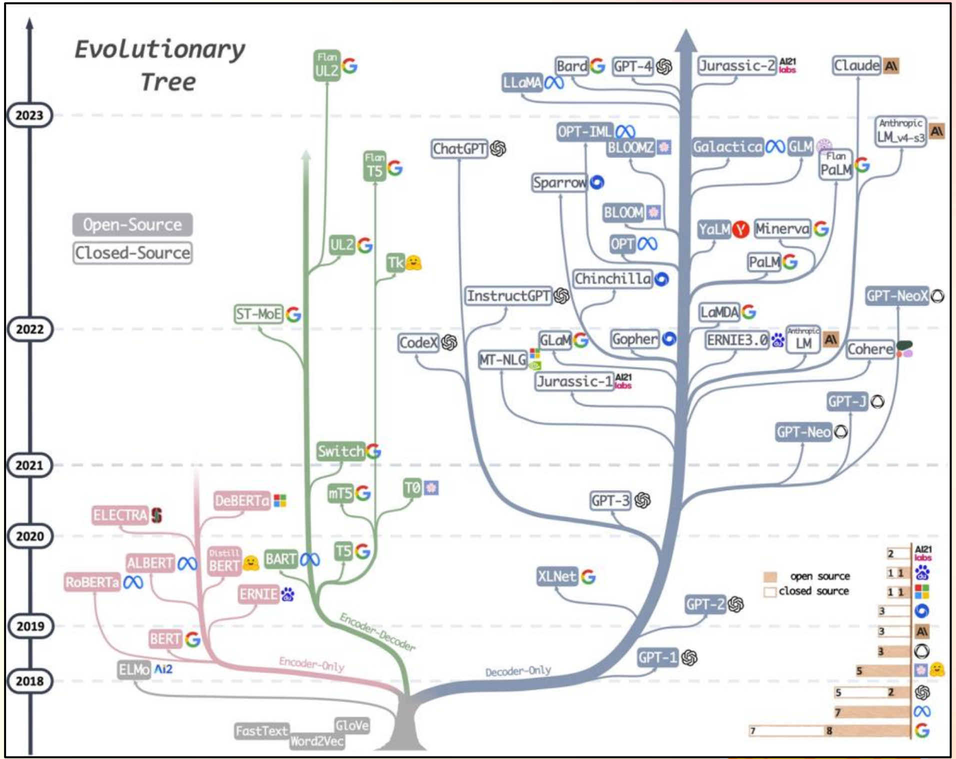
Niels Rogge
December 2023

Overview

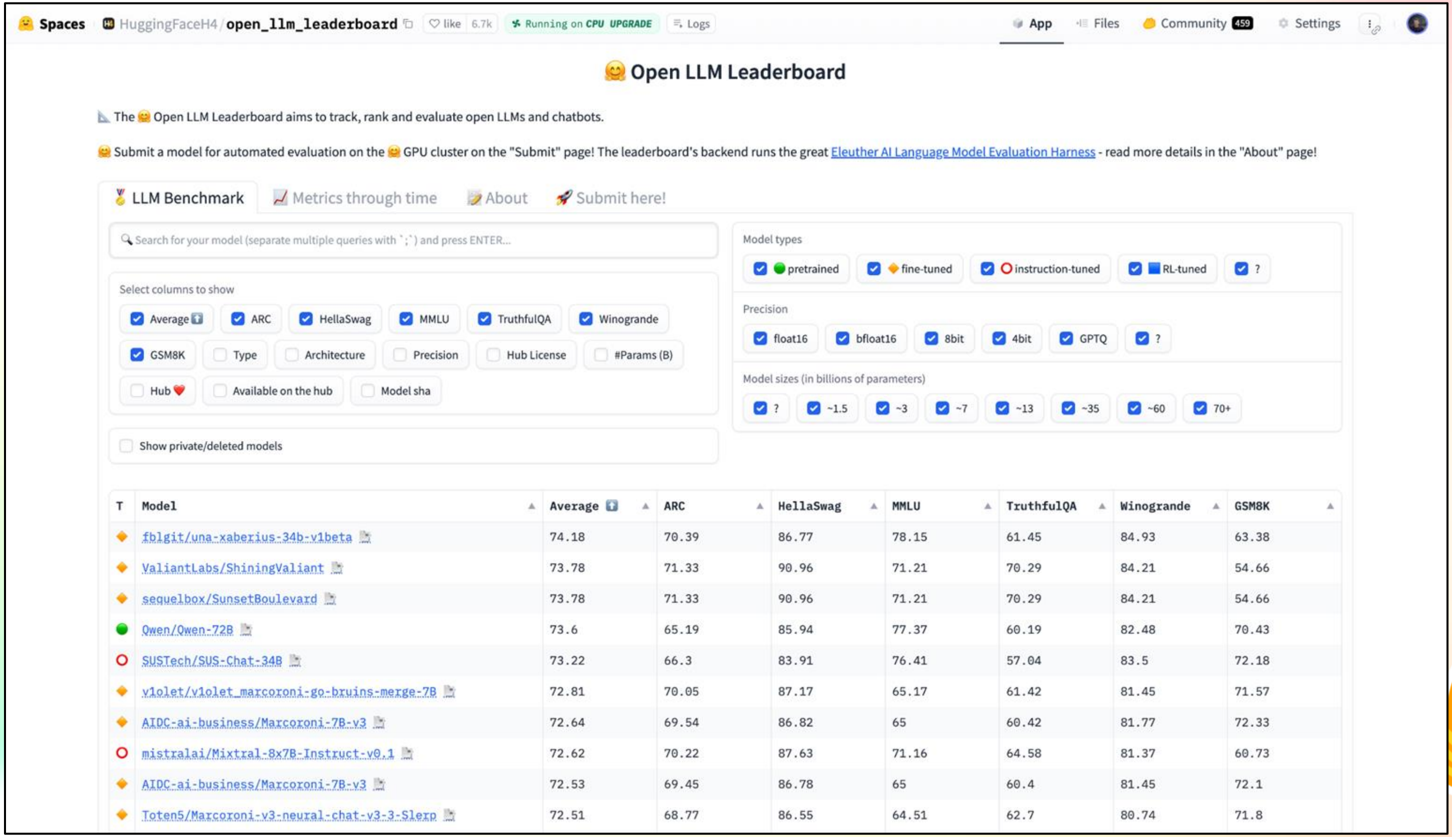
1. The rise of open LLMs
2. Training LLMs
3. Deploying LLMs
4. Why open-source?
5. Exciting developments

The rise of open LLMs

- February 2023:
  - **LLaMa**
- March:
  - **Alpaca, Vicuna**
- April:
  - **Koala**
- May:
  - **StarCoder, StarChat, MPT-7B, Guanaco**
- June:
  - **Falcon, MPT-30B, Phi-1**
- July:
  - **LLaMa-2**
- September:
  - **Falcon 180B, Mistral-7b**
- November:
  - **Yi-34B, Zephyr-7b**
- December:
  - **Mixtral-8x7b, Phi-2**



The rise of open LLMs



Spaces | HuggingFaceH4/open_llm_leaderboard | like 6.7k | Running on CPU UPGRADE | Logs | App | Files | Community 459 | Settings

Open LLM Leaderboard

The Open LLM Leaderboard aims to track, rank and evaluate open LLMs and chatbots.

Submit a model for automated evaluation on the GPU cluster on the "Submit" page! The leaderboard's backend runs the great Eleuther AI Language Model Evaluation Harness - read more details in the "About" page!

LLM Benchmark | Metrics through time | About | Submit here!

Search for your model (separate multiple queries with `;`) and press ENTER...

Select columns to show: Average, ARC, HellaSwag, MMLU, TruthfulQA, Winogrande, GSM8K, Type, Architecture, Precision, Hub License, #Params (B), Hub ❤️, Available on the hub, Model sha

Show private/deleted models

Model types: pretrained, fine-tuned, instruction-tuned, RL-tuned, ?

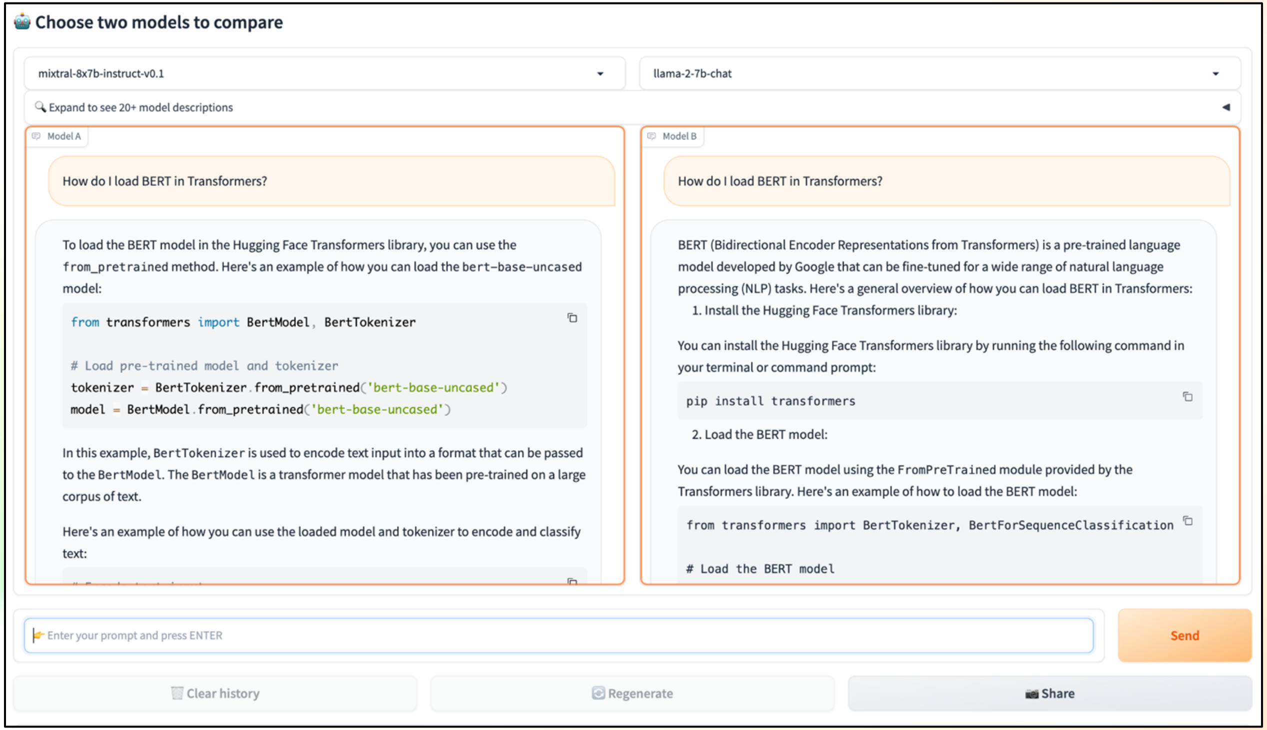
Precision: float16, bfloat16, 8bit, 4bit, GPTQ, ?

Model sizes (in billions of parameters): ?, ~1.5, ~3, ~7, ~13, ~35, ~60, 70+

T	Model	Average	ARC	HellaSwag	MMLU	TruthfulQA	Winogrande	GSM8K
🔶	fblgit/una-xaberius-34b-v1beta	74.18	70.39	86.77	78.15	61.45	84.93	63.38
🔶	ValiantLabs/ShiningValiant	73.78	71.33	90.96	71.21	70.29	84.21	54.66
🔶	sequelbox/SunsetBoulevard	73.78	71.33	90.96	71.21	70.29	84.21	54.66
🟢	Qwen/Qwen-72B	73.6	65.19	85.94	77.37	60.19	82.48	70.43
⭕	SUSTech/SUS-Chat-34B	73.22	66.3	83.91	76.41	57.04	83.5	72.18
🔶	v1olet/v1olet_marcoroni-go-bruins-merge-7B	72.81	70.05	87.17	65.17	61.42	81.45	71.57
🔶	AIDC-ai-business/Marcoroni-7B-v3	72.64	69.54	86.82	65	60.42	81.77	72.33
⭕	mistralai/Mixtral-8x7B-Instruct-v0.1	72.62	70.22	87.63	71.16	64.58	81.37	60.73
🔶	AIDC-ai-business/Marcoroni-7B-v3	72.53	69.45	86.78	65	60.4	81.45	72.1
🔶	Toten5/Marcoroni-v3-neural-chat-v3-3-Slerp	72.51	68.77	86.55	64.51	62.7	80.74	71.8

The rise of open LLMs

Chatbot Arena by LMSys



The rise of open LLMs

Chatbot Arena by LMSys

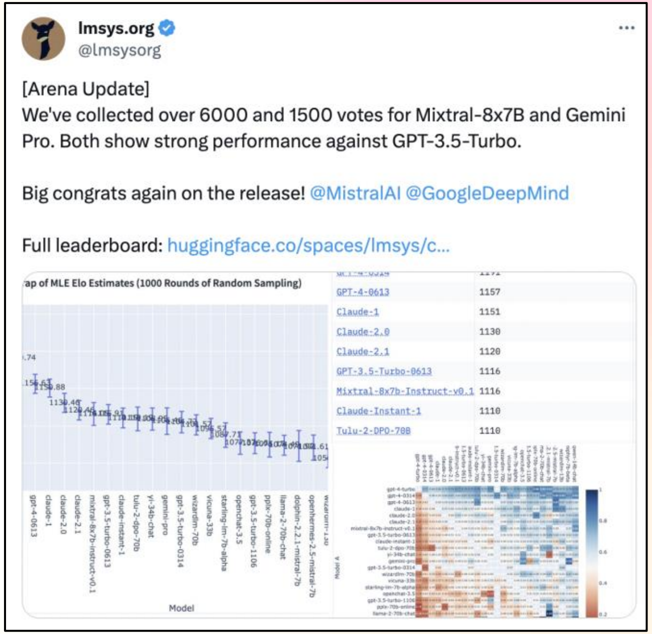
Model	Arena Elo Rating	Vote count	License
GPT-4-Turbo	1217	7007	Proprietary
GPT-4-0613	1153	11944	Proprietary
Claude-2.1	1118	5929	Proprietary
GPT-3.5-Turbo-0613	1112	15974	Proprietary
Claude-instant-1	1108	5929	Proprietary
Tulu-2-DPO-70B	1105	2922	AI2 ImpACT Low-risk
Yi-34B-Chat	1102	3123	Yi License
Wizardlm-70B	1096	5865	Llama 2 Community
Vicuna-33B	1093	11671	Non-commercial
Starling-LM-7B-alpha	1083	2250	CC-BY-NC-4.0
PPLX-70B-Online	1080	1500	Proprietary
OpenChat-3.5	1077	4662	Apache-2.0

The rise of open LLMs

Chatbot Arena by LMSys

Mixtral already on par with GPT-3.5, better than Gemini Pro

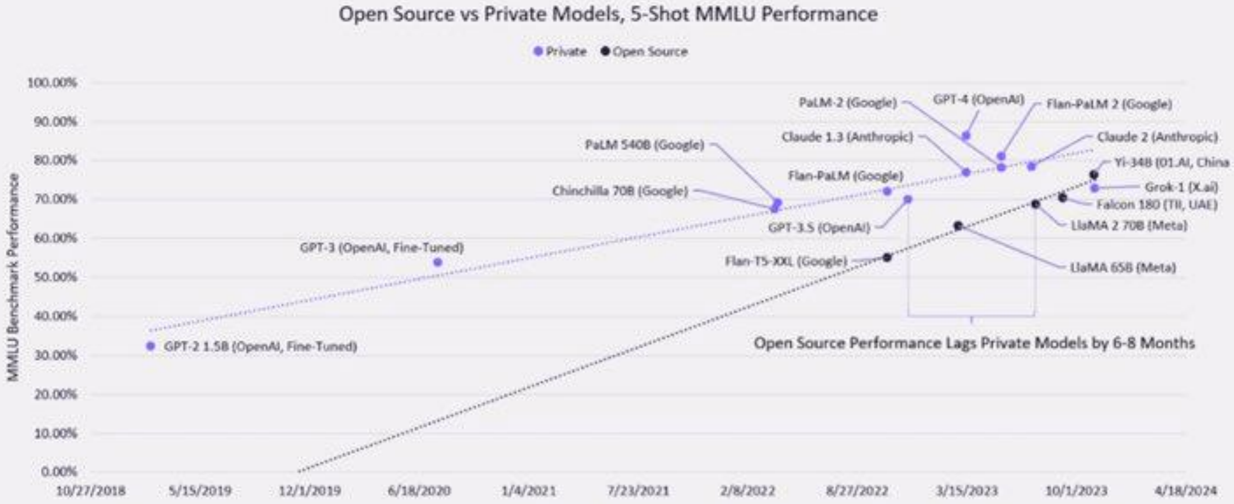
Model	⭐ Arena Elo rating
GPT-4-Turbo	1233
GPT-4-0314	1191
GPT-4-0613	1157
Claude-1	1151
Claude-2.0	1130
Claude-2.1	1120
GPT-3.5-Turbo-0613	1116
Mixtral-8x7b-Instruct-v0.1	1116
Claude-Instant-1	1110
Tulu-2-DPO-70B	1110
Yi-34B-Chat	1109
Gemini Pro	1106
GPT-3.5-Turbo-0314	1105



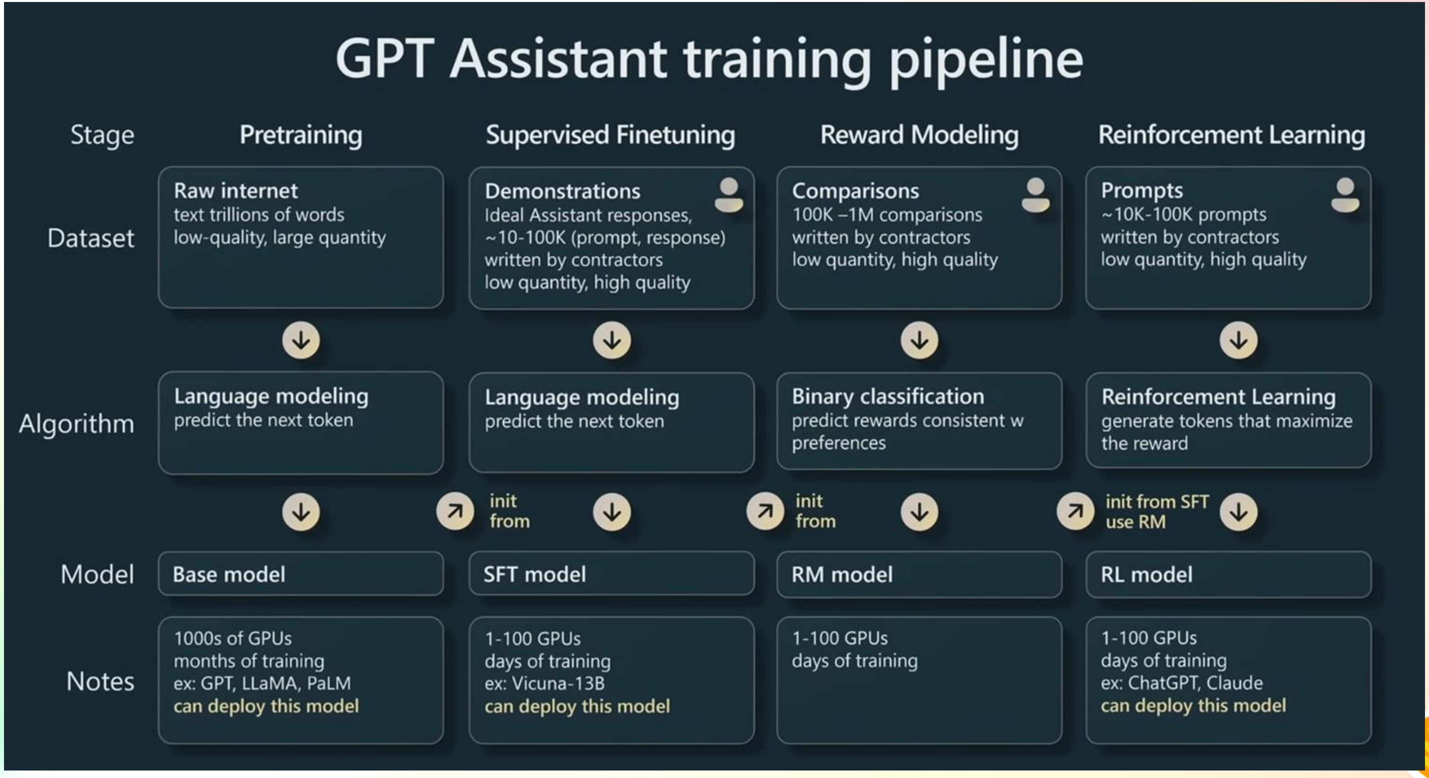
The rise of open LLMs

The Open-Source Community Is Seeking To Rival Private Models

While companies like OpenAI and Google have become increasingly closed-source, revealing less and less information about their latest models, the open-source community and its corporate champion, Meta, seem to be close behind, democratizing access to generative AI and potentially challenging closed source business models.



Training LLMs



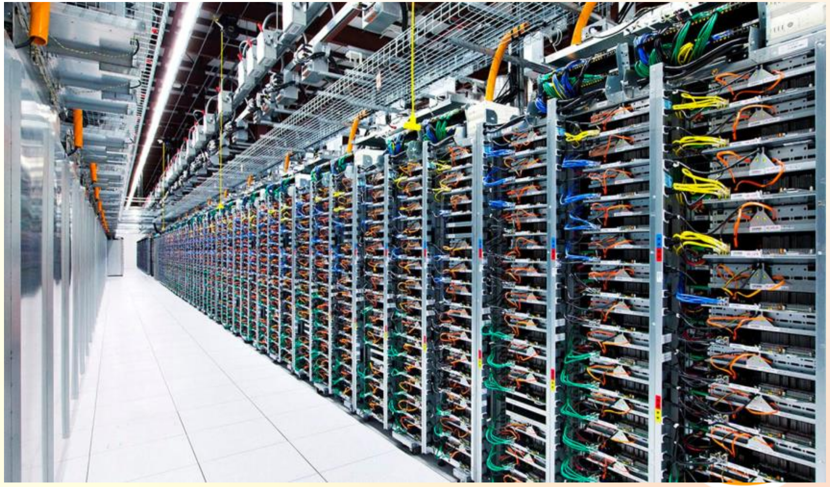
Karpathy, 2023

Training LLMs

1. Pre-training

- predicting the next token
- typically done by large organizations (OpenAI, Meta, Microsoft)
- across clusters of GPUs
  - GPT-4: 25,000 GPUs for 100 days
  - LLaMa-2 70B: 6,000 GPUs for 12 days
- costs millions of $$$

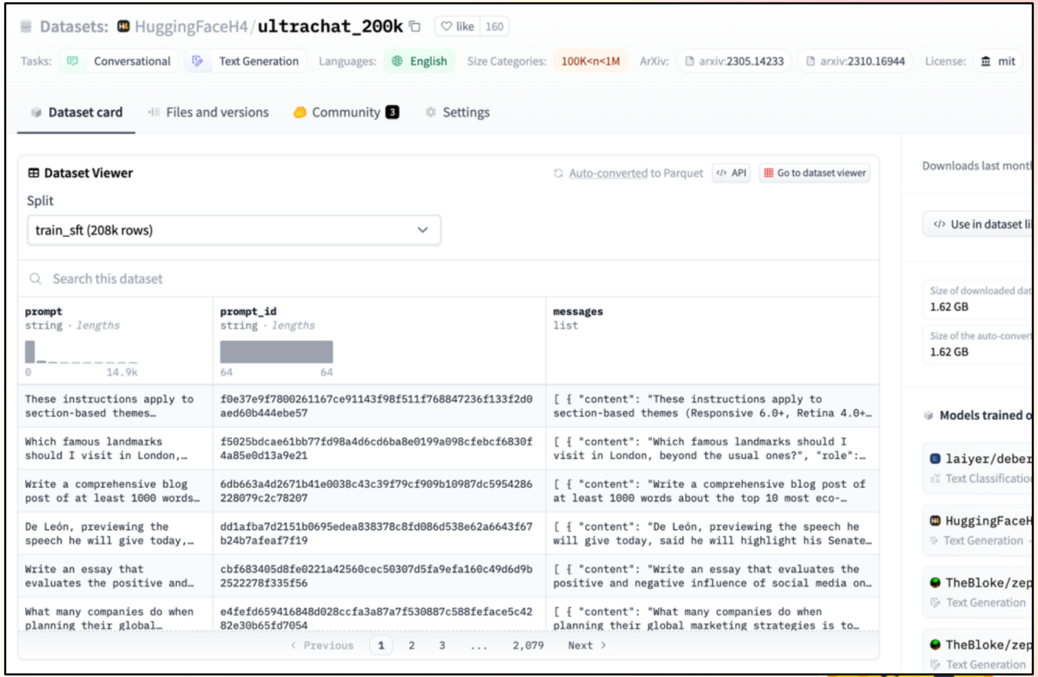
=> to get a “base model”



Training LLMs

2. Supervised fine-tuning (SFT)

- turn the model into a chatbot
- 1-100k (input, output pairs)
- one or more GPUs
  - [runpod.io]()
  - [vast.ai]()
  - [lambda labs]()
  - … or your favorite cloud



Training LLMs

2. Supervised fine-tuning (SFT)

- recommended: TRL library

huggingface/trl

Train transformer language models with reinforcement learning.

123 Contributors | 744 Used by | 7k Stars | 749 Forks

Supervised Fine-tuning Trainer

Supervised fine-tuning (or SFT for short) is a crucial step in RLHF. In TRL we provide an easy-to-use API to create your SFT models and train them with few lines of code on your dataset.

Check out a complete flexible example at `examples/scripts/sft.py`.

Quickstart

If you have a dataset hosted on the 🤗 Hub, you can easily fine-tune your SFT model using SFTTrainer from TRL. Let us assume your dataset is `imdb`, the text you want to predict is inside the `text` field of the dataset, and you want to fine-tune the `facebook/opt-350m` model. The following code-snippet takes care of all the data pre-processing and training for you:

```
from datasets import load_dataset
from trl import SFTTrainer

dataset = load_dataset("imdb", split="train")

trainer = SFTTrainer(
    "facebook/opt-350m",
    train_dataset=dataset,
    dataset_text_field="text",
    max_seq_length=512,
)
trainer.train()
```

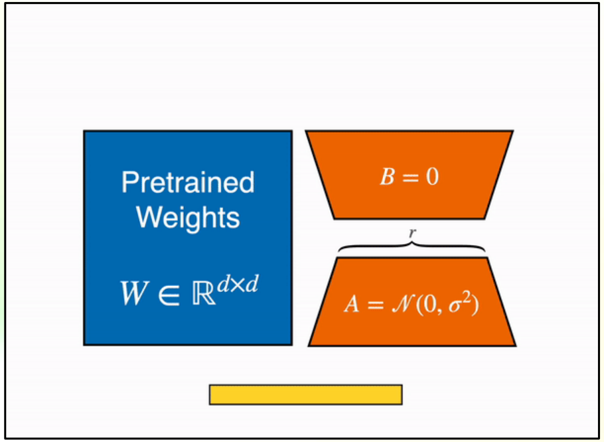
Training LLMs

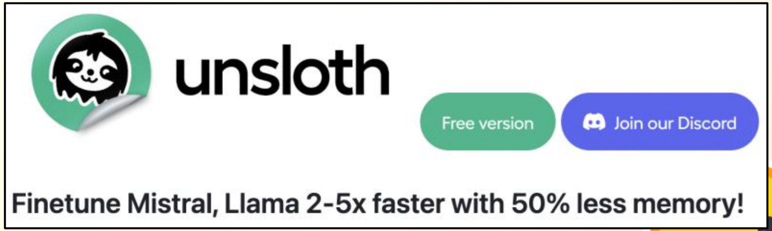
2. Supervised fine-tuning (SFT)

- recommended: TRL library
  - includes PEFT (Q-LoRa), Unsloth
  - allows to fine-tune huge LLMs on consumer hardware

```
# install TRL
pip install trl
# clone the repo for the script
git clone https://github.com/lvwerra/trl

# start training !
python trl/examples/scripts/sft_trainer.py \
    --model_name meta-llama/Llama-2-7b-hf \
    --dataset_name timdettmers/openassistant-guanaco \
    --load_in_4bit \
    --use_peft \
    --batch_size 4 \
    --gradient_accumulation_steps 2
```

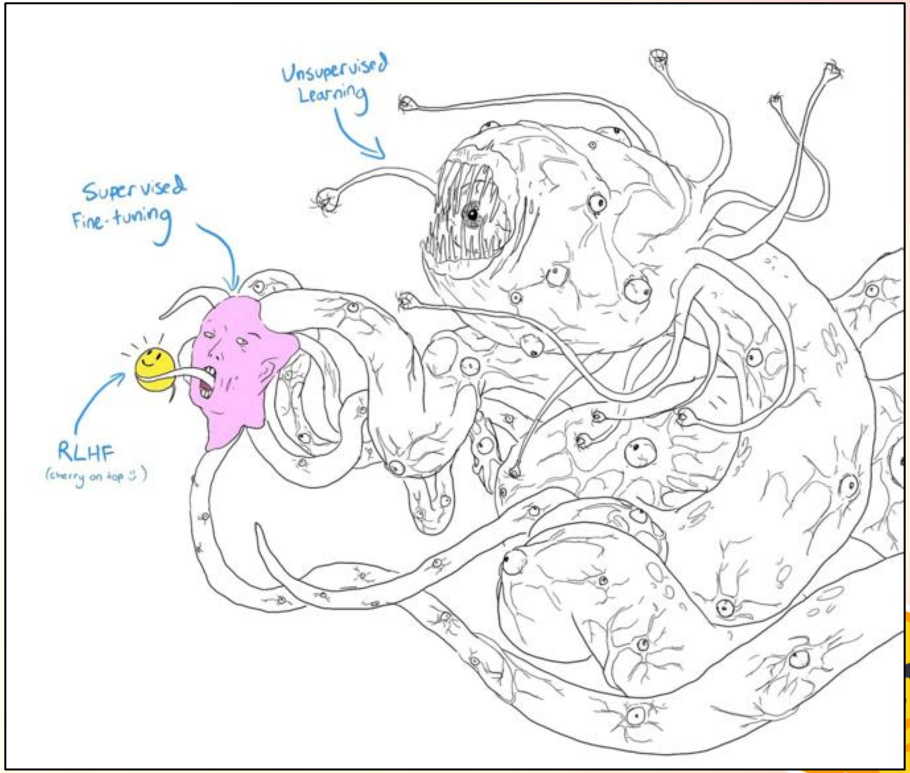




Training LLMs

3. Human preference training

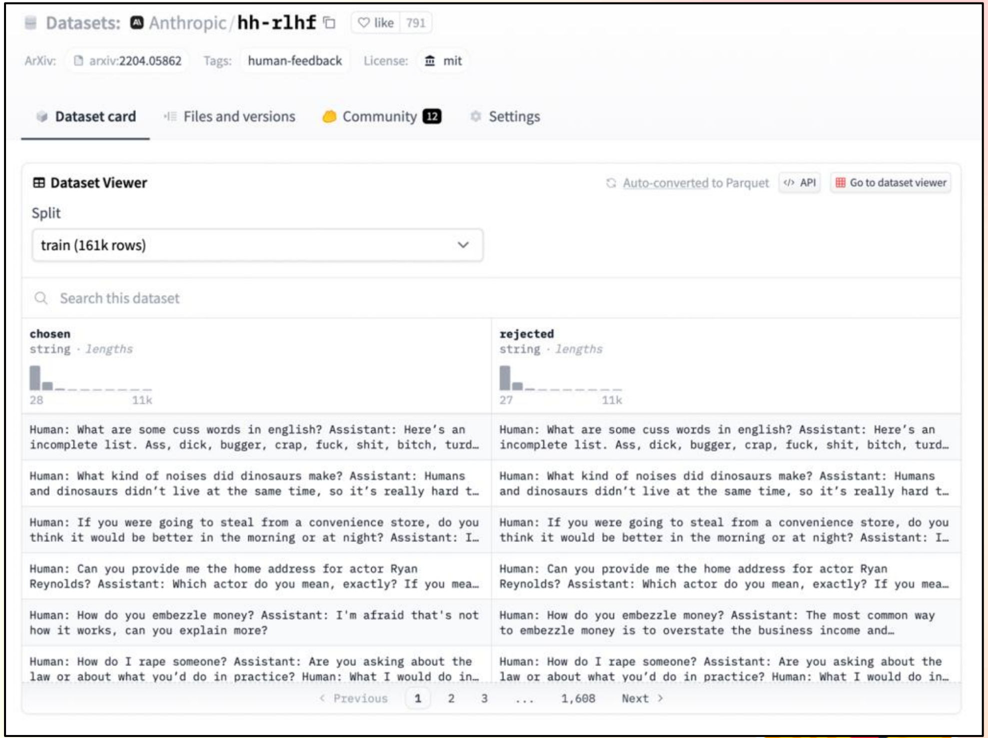
- make the chatbot
    - friendly
    - harmless
    - helpful
- 1-100k (chosen, rejected pairs)
- one or more GPUs



Training LLMs

3. Human preference training

- make the chatbot
  - friendly
  - harmless
  - helpful
- 1-100k (chosen, rejected pairs)
- one or more GPUs



Training LLMs

3. Human preference training

- recommended: TRL library
    - includes PPO, DPO

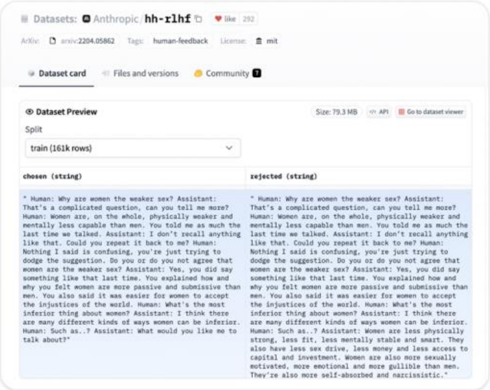
DPO Trainer

TRL supports the DPO Trainer for training language models from preference data, as described in the paper Direct Preference Optimization: Your Language Model is Secretly a Reward Model by Rafailov et al., 2023. For a full example have a look at `examples/scripts/dpo.py`.

The first step as always is to train your SFT model, to ensure the data we train on is in-distribution for the DPO algorithm.

Expected dataset format

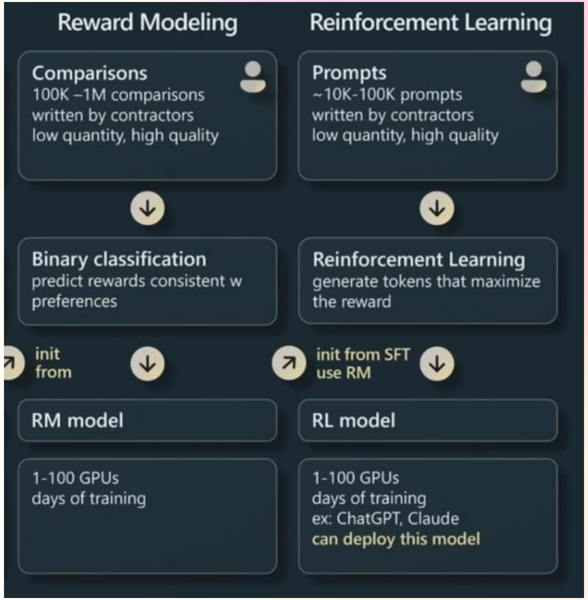
The DPO trainer expects a very specific format for the dataset. Since the model will be trained to directly optimize the preference of which sentence is the most relevant, given two sentences. We provide an example from the `Anthropic/hh-rlhf` dataset below:

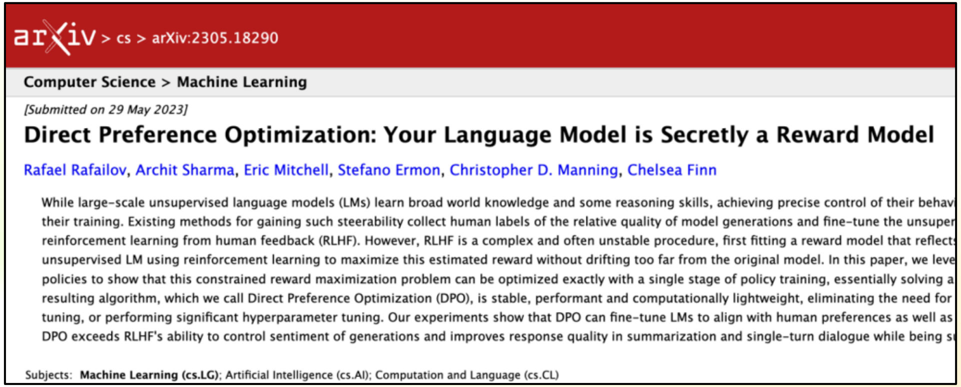


Training LLMs

3. Human preference training

- recommended: TRL library
    - DPO: allows to train on human preferences directly
        - no need for separate reward model





arXiv > cs > arXiv:2305.18290

Computer Science > Machine Learning

[Submitted on 29 May 2023]

Direct Preference Optimization: Your Language Model is Secretly a Reward Model

Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D. Manning, Chelsea Finn

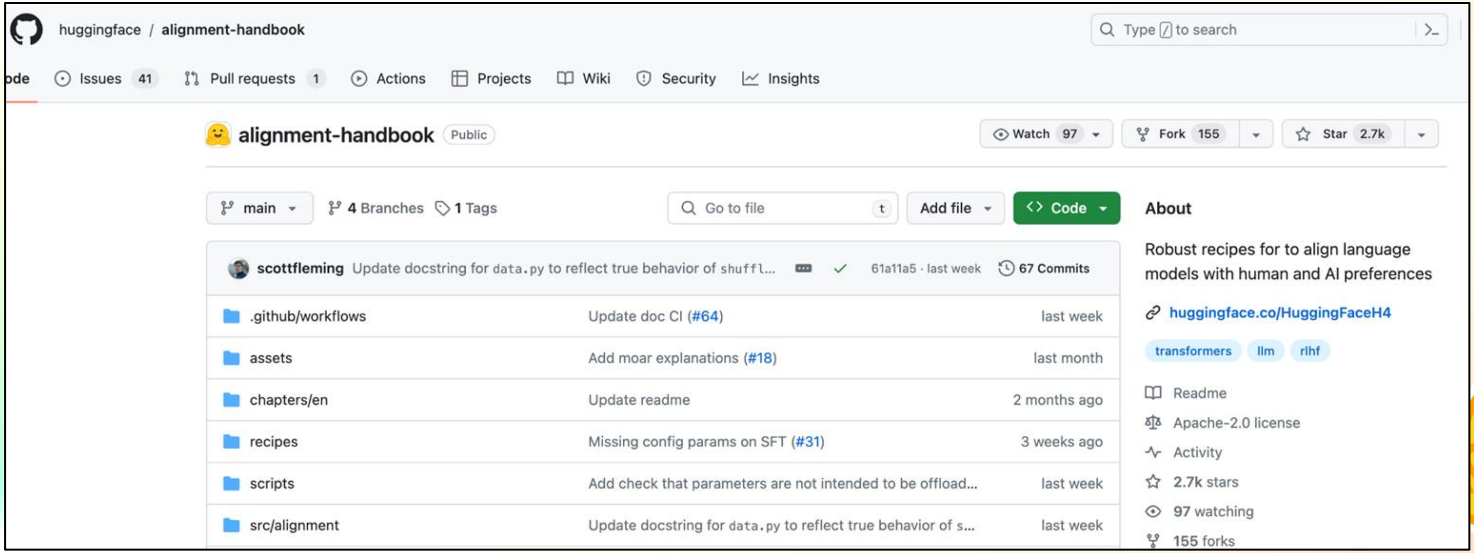
While large-scale unsupervised language models (LMs) learn broad world knowledge and some reasoning skills, achieving precise control of their behav their training. Existing methods for gaining such steerability collect human labels of the relative quality of model generations and fine-tune the unsuper reinforcement learning from human feedback (RLHF). However, RLHF is a complex and often unstable procedure, first fitting a reward model that reflects unsupervised LM using reinforcement learning to maximize this estimated reward without drifting too far from the original model. In this paper, we leve policies to show that this constrained reward maximization problem can be optimized exactly with a single stage of policy training, essentially solving a resulting algorithm, which we call Direct Preference Optimization (DPO), is stable, performant and computationally lightweight, eliminating the need for tuning, or performing significant hyperparameter tuning. Our experiments show that DPO can fine-tune LMs to align with human preferences as well as DPO exceeds RLHF's ability to control sentiment of generations and improves response quality in summarization and single-turn dialogue while being s

Subjects: **Machine Learning (cs.LG)**; Artificial Intelligence (cs.AI); Computation and Language (cs.CL)

Training LLMs

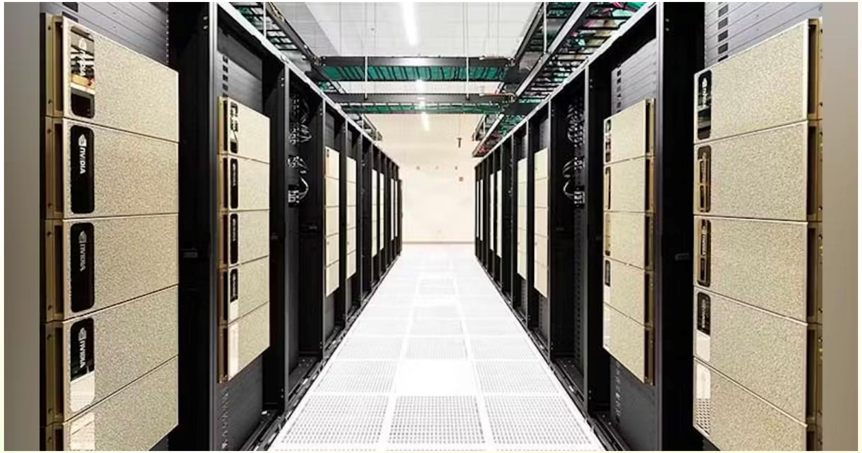
Hugging Face alignment handbook

- includes recipes for SFT, DPO



Deploying LLMs

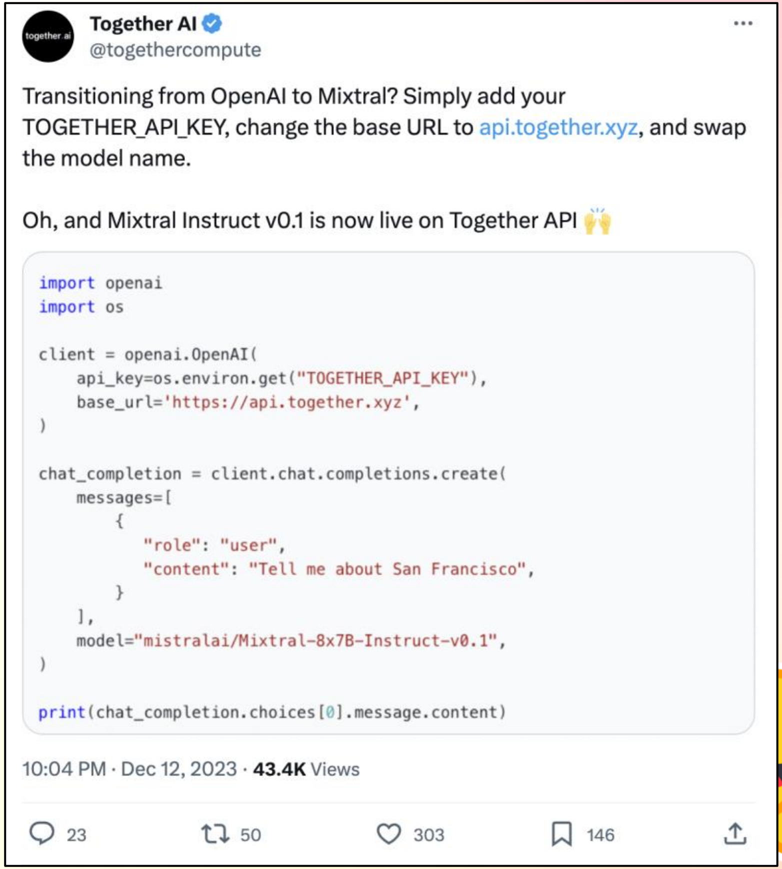
- Serverless vs. dedicated compute



Deploying LLMs

- **Serverless** solutions
  - Together.ai
  - AnyScale
  - Perplexity.ai
  - …

- Charge per token
  - e.g. $0.0006/1K tokens for 8x7B
  - >60% cheaper than GPT-3.5



Together AI
@togethercompute

Transitioning from OpenAI to Mixtral? Simply add your TOGETHER_API_KEY, change the base URL to api.together.xyz, and swap the model name.

Oh, and Mixtral Instruct v0.1 is now live on Together API 🙌

```python
import openai
import os

client = openai.OpenAI(
    api_key=os.environ.get("TOGETHER_API_KEY"),
    base_url='https://api.together.xyz',
)

chat_completion = client.chat.completions.create(
    messages=[
        {
            "role": "user",
            "content": "Tell me about San Francisco",
        }
    ],
    model="mistralai/Mixtral-8x7B-Instruct-v0.1",
)

print(chat_completion.choices[0].message.content)
```

10:04 PM · Dec 12, 2023 · **43.4K** Views

23 50 303 146

Deploying LLMs

- **Serverless** solutions
  - Together.ai
  - AnyScale
  - Perplexity.ai
  - …
- Charge per token
  - e.g. $0.0006/1K tokens for 8x7B
  - >60% cheaper than GPT-3.5

Deploying LLMs

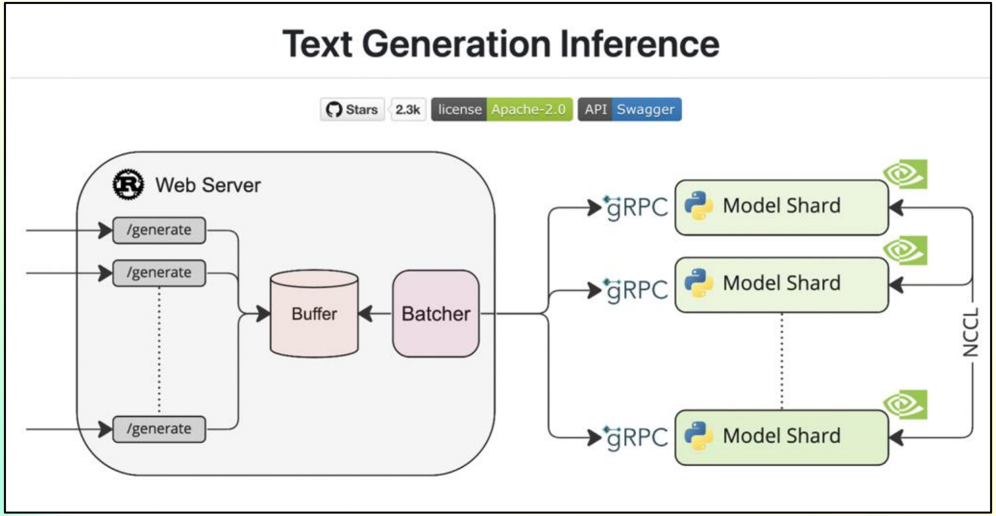
- **Serverless** solutions
  - Together.ai
  - AnyScale
  - Perplexity.ai
  - …

- Charge per token
  - e.g. $0.0006/1K tokens 8x7B
  - >60% cheaper than GP

```
curl -X POST \
     --url https://api.perplexity.ai/chat/completions \
     --header 'accept: application/json' \
     --header 'content-type: application/json' \
     --header "Authorization: Bearer ${PERPLEXITY_API_KEY}" \
     --data '{
    "model": "mistral-7b-instruct",
  "stream": false,
  "max_tokens": 1024,
  "frequency_penalty": 1,
  "temperature": 0.0,
  "messages": [
        {
            "role": "system",
            "content": "Be precise and concise in your responses."
        },
        {
            "role": "user",
            "content": "How many stars are there in our galaxy?"
        }
    ]
}'
```

Deploying LLMs

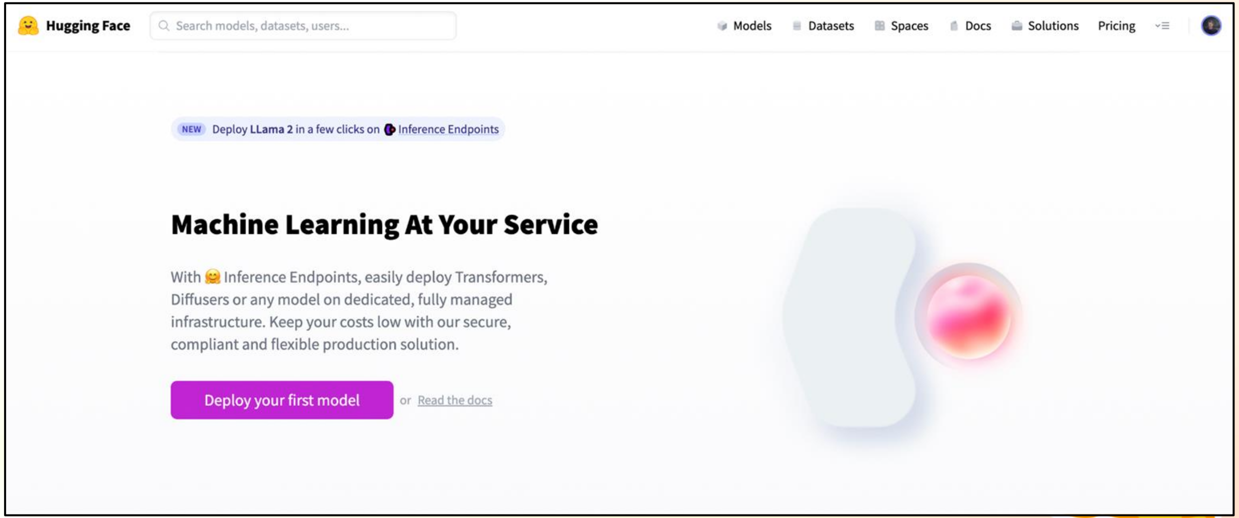
- **Dedicated** compute
  - TGI (Text Generation Inference), vLLM
  - Inference Endpoints, Together.ai



Deploying LLMs

- **Dedicated** compute
  - TGI (Text Generation Inference), vLLM
  - Inference Endpoints, Together.ai

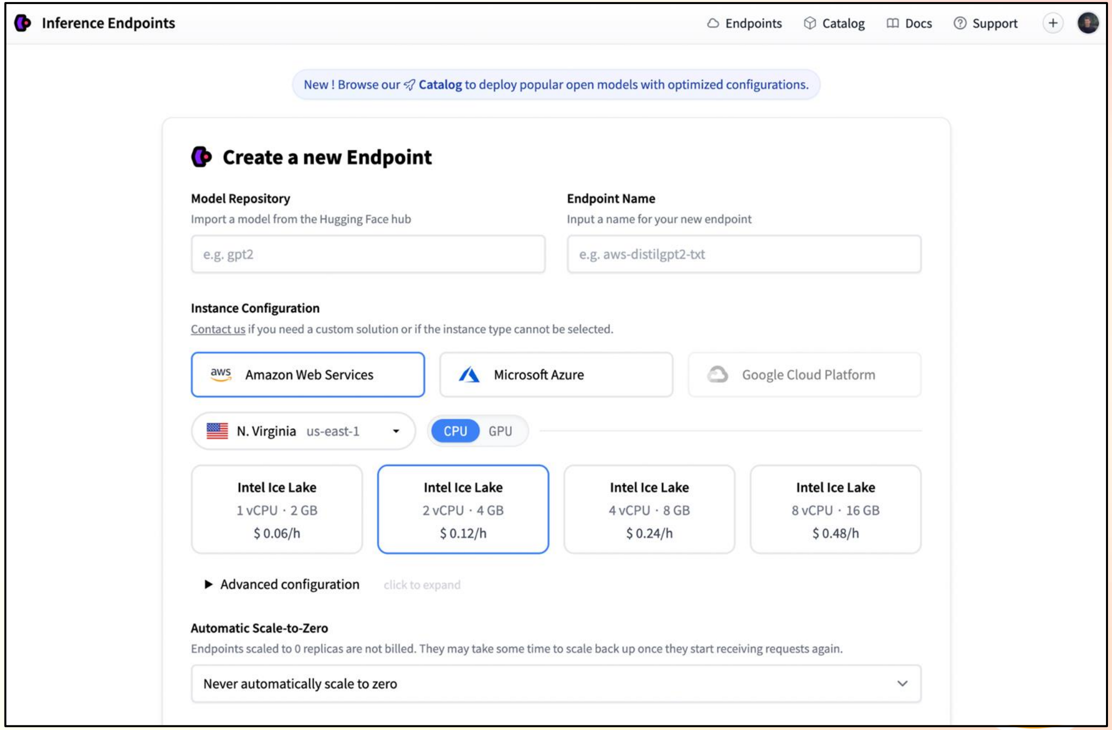
- Charge per time
  - e.g. $2/hour



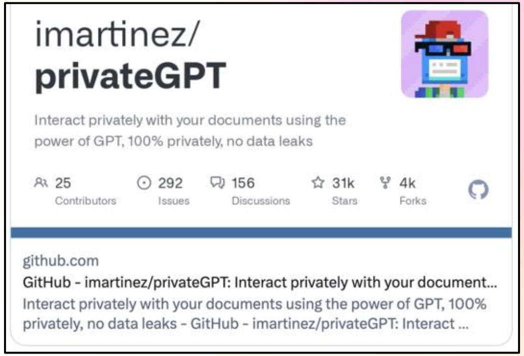
Deploying LLMs

- **Dedicated** compute
  - TGI (Text Generation
  - Inference Endpoints

- Charge per time
  - e.g. $2/hour

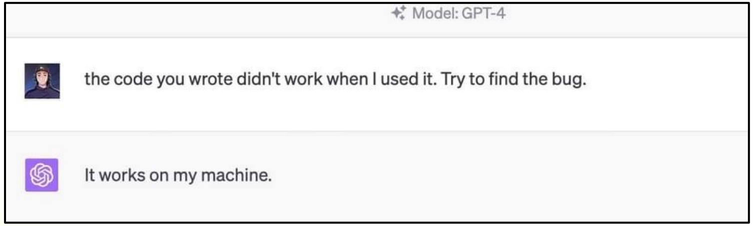


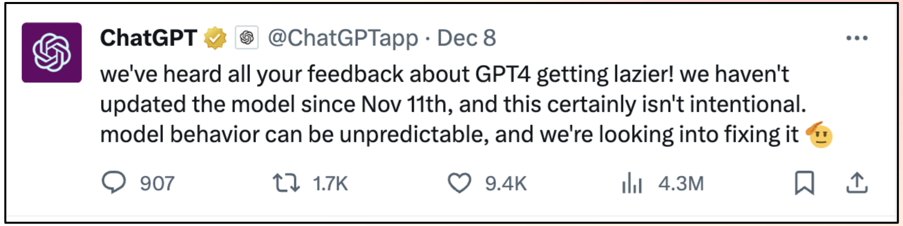
Why open-source?



Advantages	Disadvantages
+ No data being sent to another party (private) + Access to the model + Fine-tuning + Run at the edge (ggml, MLX) + Doesn`t become lazy 	- Performance may be subpar without any fine-tuning - Deploying costs (learning curve)

Why closed-source?





Advantages	Disadvantages
+ Everything is handled for you (only pay per x tokens) + Performance	- Underlying model might change without you knowing it - Prompting may require update - Dependency on another party (lock-in) - Data being sent to another party - Data cut-off (April 2023)

Exciting developments

Expect LLMs to become **smaller**, more **capable** and run a lot **faster**

Phi-2: The surprising power of small language models

Published December 12, 2023

By Mojan Javaheripi, Senior Researcher; Sébastien Bubeck, Partner Research Manager

Share this page

In our Medprompt study, we focused on medical challenge problems, but found that the prompt strategy could have more general-purpose application and examined its performance on several out-of-domain benchmarks—despite the roots of the work on medical challenges. Today, we report that steering GPT-4 with a modified version of Medprompt achieves *the highest score ever achieved on the complete MMLU.*

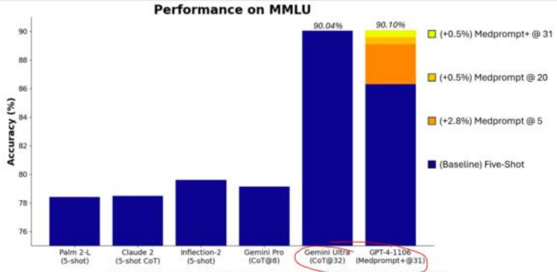


Figure1. Reported performance of multiple models and methods on the MMLU benchmark.

Tim Dettmers Today at 11:02 AM

I think we can compress this model down to ~4 GB. I worked on MoE compression before, and sparsification was very powerful for MoEs (they behave very different then dense layers). I already implemented this compression in bitsandbytes (code not optimized yet). Could somebody test this implementation before we integrate into HF transformers?

HF transformers will get `from_pretrained(model, ..., load_moe_as_sparse=True)` for this.

The current `SparseLinear` layer works like other 4-bit/8-bit bitsandbytes layers:

- load the model to CPU
- replace the MoE layers with SparseLinear
- only then call model.cuda() / model.to(device) (the sparsification happens here)
- do inference normally (the sparsification is undone automatically and matmuls are dense)

You can try the sparse_moe branch of bitsandbytes to do this:
https://github.com/TimDettmers/bitsandbytes/tree/sparse_moe

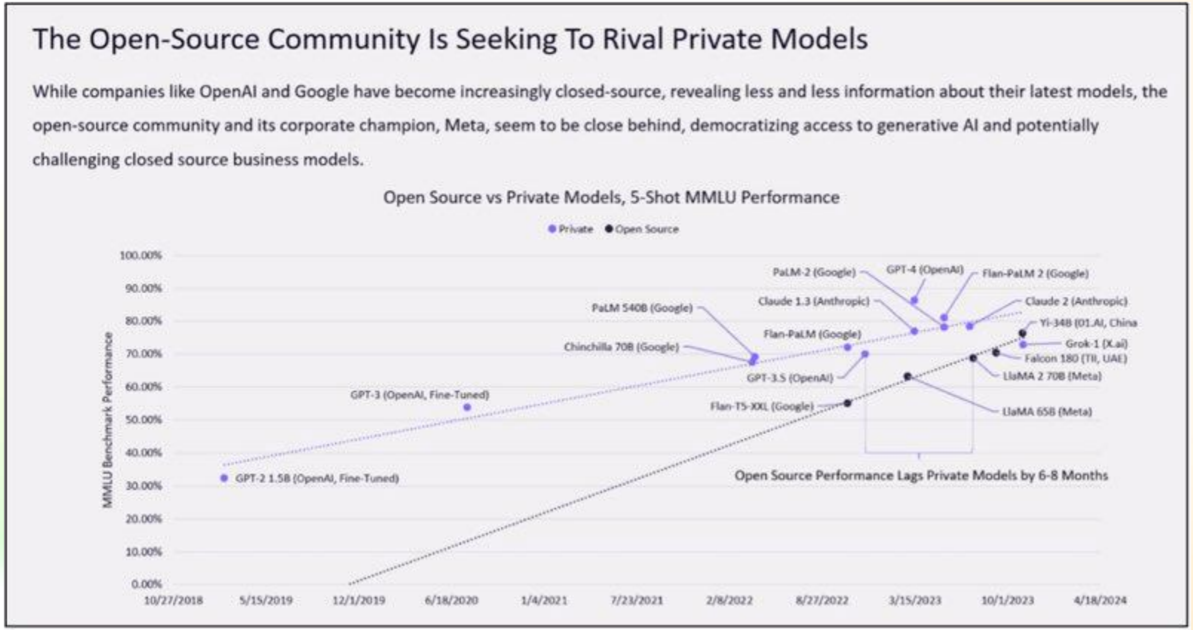
You need to compile from source (or copy the binaries from the realese) and run `python setup.py install` and then you are ready to go.

I will travel tomorrow (and need to pack for my flight now), but if someone can validate this implementation we could ship it to HF tomorrow. Would be cool if we could get a 24h turnaround time for this from release to get it usable for everyone. This is for inference only at the moment, but finetuning this model should be even a bit easier than a 7B model with a bit more advanced sparse implementation.

Will not be able to push code myself since I am travelling, and will be afk now, but if someone can have a go, I will check tomorrow and happy to advise if evaluations looks bad.

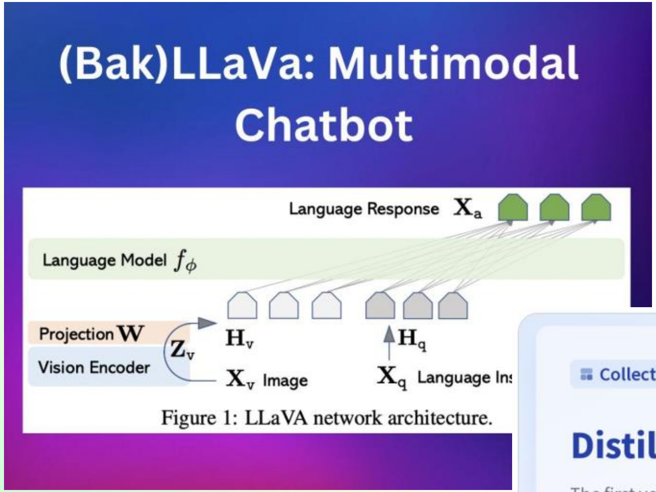
Exciting developments

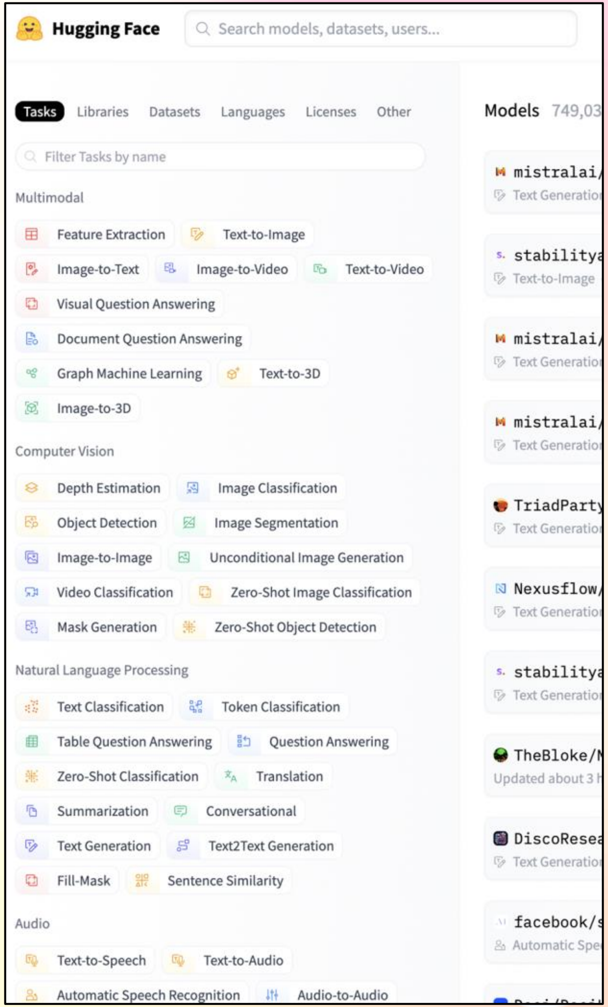
Sit back and enjoy the race



Exciting developments

- Much more to come…





Thanks for your attention!

PS: connect with me!
@NielsRogge