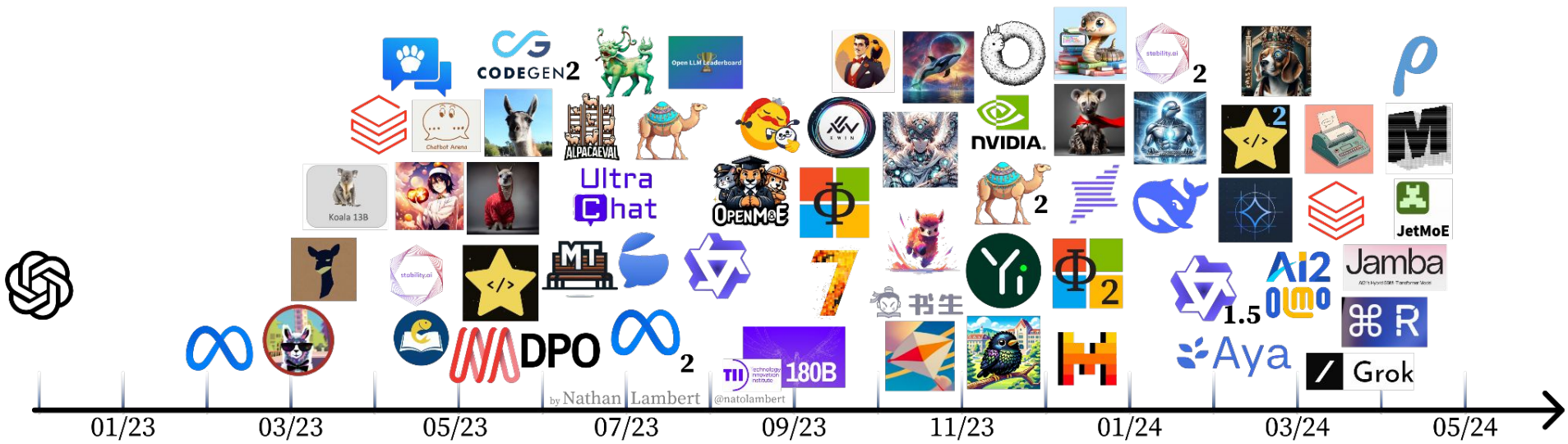
Aligning open language models

Nathan Lambert || Allen Institute for AI || @natolambert

Stanford CS25: Transformers United V4



Make them more useful for our tasks

A heavily abbreviated history of language models (LMs)

A heavily abbreviated history of LMs

1948: Claude Shannon models English

Approx range of chars to Create an LLM

Built on auto-regressive loss function

3. THE SERIES OF APPROXIMATIONS TO ENGLISH

To give a visual idea of how this series of processes approaches a language, typical sequences in the approximations to English have been constructed and are given below. In all cases we have assumed a 27-symbol "alphabet," the 26 letters and a space.

1. Zero-order approximation (symbols independent and equiprobable).

   XFOML RXKHRJFFJUJ ZLPWCFWKCYJ FFJEYVKCQSGHYD QPAAMKBZAACIBZL-HJQD.

2. First-order approximation (symbols independent but with frequencies of English text).

   OCRO HLI RGWR NMIELWIS EU LL NBNESEBYA TH EEI ALHENHTTPA OOBTTVA NAH BRL.

3. Second-order approximation (digram structure as in English).

   ON IE ANTSOUTINYS ARE T INCTORE ST BE S DEAMY ACHIN D ILONASIVE TU-COOWE AT TEASONARE FUSO TIZIN ANDY TOBE SEACE CTISBE.

4. Third-order approximation (trigram structure as in English).

   IN NO IST LAT WHEY CRATICT FROURE BIRS GROCID PONDENOME OF DEMONS-TURES OF THE REPTAGIN IS REGOACTIONA OF CRE.

5. First-order word approximation. Rather than continue with tetragram, . . . , *n*-gram structure it is easier and better to jump at this point to word units. Here words are chosen independently but with their appropriate frequencies.

   REPRESENTING AND SPEEDILY IS AN GOOD APT OR COME CAN DIFFERENT NAT-URAL HERE HE THE A IN CAME THE TO OF TO EXPERT GRAY COME TO FURNISHES THE LINE MESSAGE HAD BE THESE.

6. Second-order word approximation. The word transition probabilities are correct but no further structure is included.

   THE HEAD AND IN FRONTAL ATTACK ON AN ENGLISH WRITER THAT THE CHAR-ACTER OF THIS POINT IS THEREFORE ANOTHER METHOD FOR THE LETTERS THAT THE TIME OF WHO EVER TOLD THE PROBLEM FOR AN UNEXPECTED.

Shannon 1948

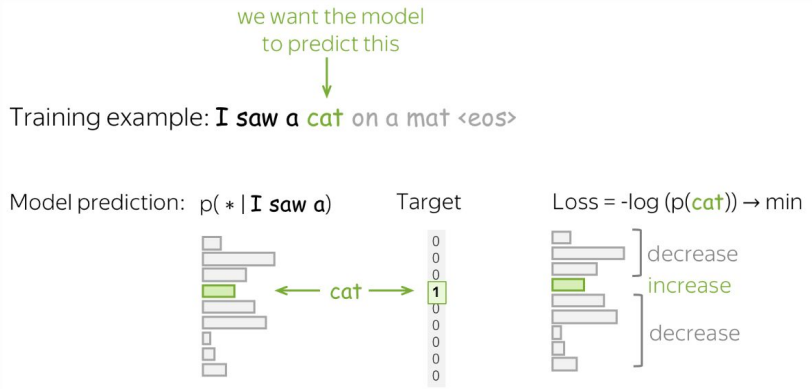
A heavily abbreviated history of LMs

1948: Claude Shannon models English

1948-2017: 🤯

$$Loss(p^*, p) = -\log(p_{y_t}) = -\log(p(y_t|y_{<t})).$$

At each step, we maximize the probability a model assigns to the correct token. Look at the illustration for a single timestep.

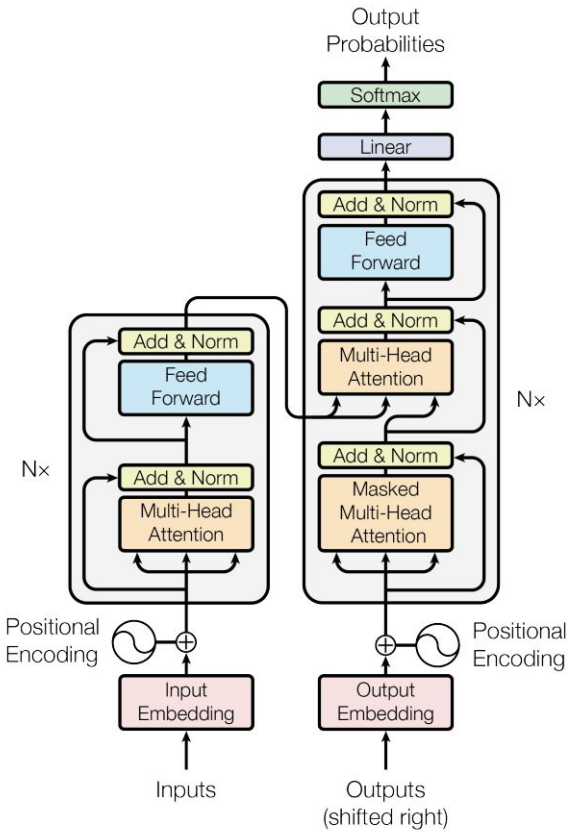


A heavily abbreviated history of LMs

1948: Claude Shannon models English

1948-2017: 🤯

2017: the transformer is born



Vaswani et al. 2017

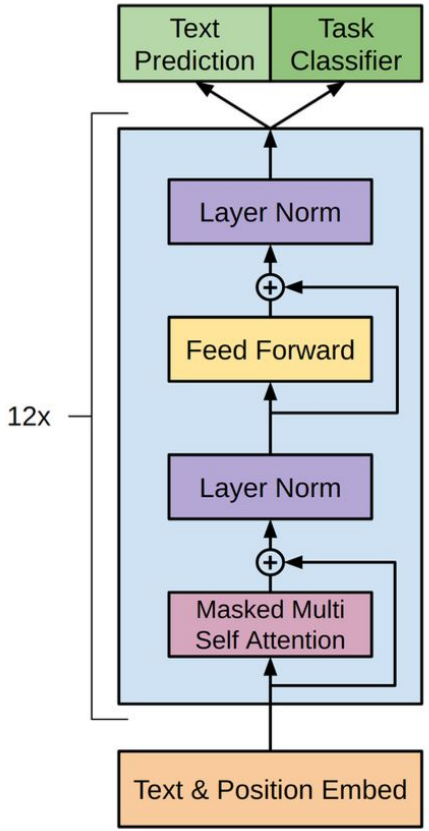
A heavily abbreviated history of LMs

1948: Claude Shannon models English

1948-2017: 🤯

2017: the transformer is born

2018: GPT-1, ELMo, and BERT released RADICALI Time



Radford et al. 2018, Devlin et al. 2018

A heavily abbreviated history of LMs

1948: Claude Shannon models English

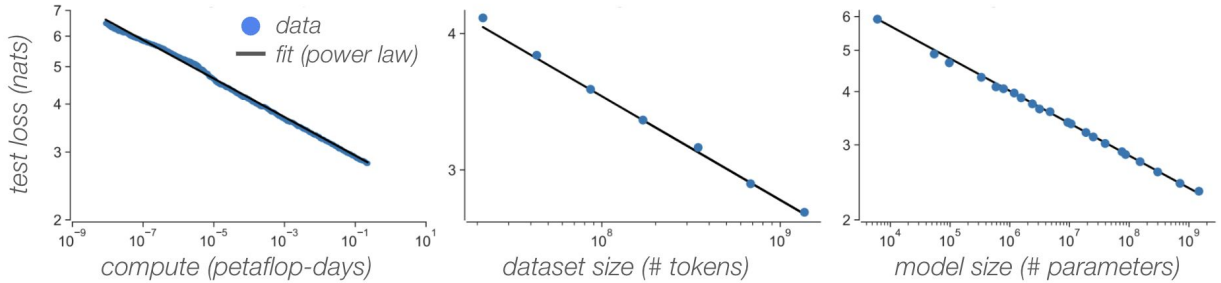
1948-2017: 🤯

2017: the transformer is born

2018: GPT-1, ELMo and BERT released

2019: GPT-2 and scaling laws

Radford et al. 2019, Kaplan et al. 2020



A heavily abbreviated history of LMs

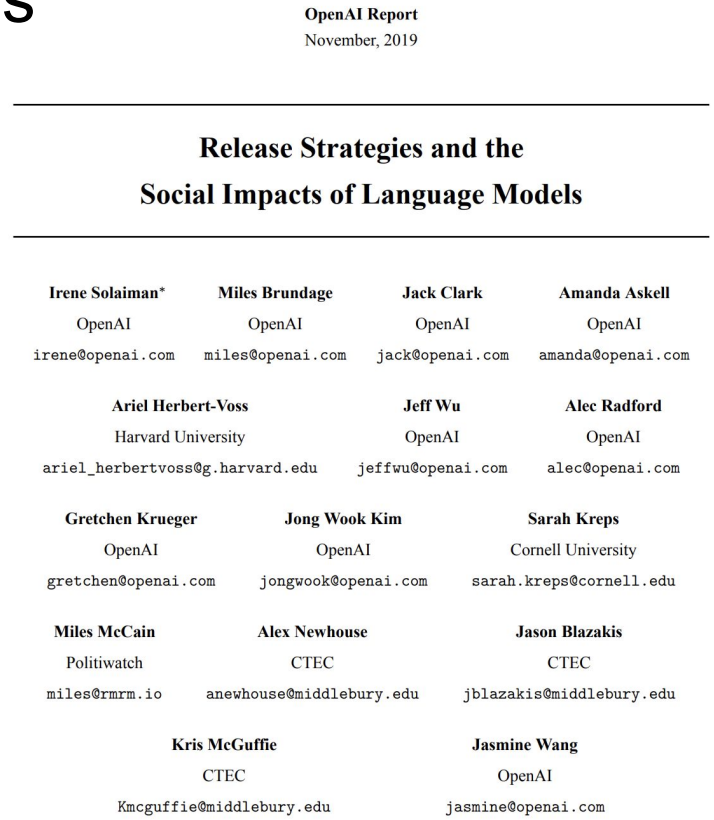
1948: Claude Shannon models English

1948-2017: 🤯

2017: the transformer is born

2018: GPT-1, ELMo and BERT released

2019: GPT-2 and scaling laws (and releases)



OpenAI Report
November, 2019

Release Strategies and the Social Impacts of Language Models

Irene Solaiman*
OpenAI
irene@openai.com

Miles Brundage
OpenAI
miles@openai.com

Jack Clark
OpenAI
jack@openai.com

Amanda Askell
OpenAI
amanda@openai.com

Ariel Herbert-Voss
Harvard University
ariel_herbertvoss@g.harvard.edu

Jeff Wu
OpenAI
jeffwu@openai.com

Alec Radford
OpenAI
alec@openai.com

Gretchen Krueger
OpenAI
gretchen@openai.com

Jong Wook Kim
OpenAI
jongwook@openai.com

Sarah Kreps
Cornell University
sarah.kreps@cornell.edu

Miles McCain
Politiwatch
miles@rmrm.io

Alex Newhouse
CTEC
anewhouse@middlebury.edu

Jason Blazakis
CTEC
jblazakis@middlebury.edu

Kris McGuffie
CTEC
Kmcguffie@middlebury.edu

Jasmine Wang
OpenAI
jasmine@openai.com

A heavily abbreviated history of LMs

1948: Claude Shannon models English

1948-2017: 🤯

2017: the transformer is born

2018: GPT-1, ELMo and BERT released

2019: GPT-2 and scaling laws

2020: GPT-3 surprising capabilities. many harms

Zero-shot

The model predicts the answer given only a natural language description of the task. No gradient updates are performed.

```
Translate English to French:    ← task description
cheese =>                       ← prompt
```

One-shot

In addition to the task description, the model sees a single example of the task. No gradient updates are performed.

```
Translate English to French:    ← task description
sea otter => loutre de mer      ← example
cheese =>                       ← prompt
```

Few-shot

In addition to the task description, the model sees a few examples of the task. No gradient updates are performed.

```
Translate English to French:    ← task description
sea otter => loutre de mer      ← examples
peppermint => menthe poivrée    ←
plush girafe => girafe peluche  ←
cheese =>                       ← prompt
```

A heavily abbreviated history of LMs

1948: Claude Shannon models English

1948-2017: 🤯

2017: the transformer is born

2018: GPT-1, ELMo and BERT released

2019: GPT-2 and scaling laws

2020: GPT-3 surprising capabilities

2021: Stochastic parrots

On the Dangers of Stochastic Parrots: Can Language Models Be Too Big? 🦜

Emily M. Bender*
ebender@uw.edu
University of Washington
Seattle, WA, USA

Timnit Gebru*
timnit@blackinai.org
Black in AI
Palo Alto, CA, USA

Angelina McMillan-Major
aymm@uw.edu
University of Washington
Seattle, WA, USA

Shmargaret Shmitchell
shmargaret.shmitchell@gmail.com
The Aether

ABSTRACT

The past 3 years of work in NLP have been characterized by the development and deployment of ever larger language models, especially for English. BERT, its variants, GPT-2/3, and others, most recently Switch-C, have pushed the boundaries of the possible both through architectural innovations and through sheer size. Using these pretrained models and the methodology of fine-tuning them for specific tasks, researchers have extended the state of the art on a wide array of tasks as measured by leaderboards on specific benchmarks for English. In this paper, we take a step back and ask: How big is too big? What are the possible risks associated with this technology and what paths are available for mitigating those risks? We provide recommendations including weighing the environmental and financial costs first, investing resources into curating and carefully documenting datasets rather than ingesting everything on the web, carrying out pre-development exercises evaluating how the planned approach fits into research and development goals and supports stakeholder values, and encouraging research directions beyond ever larger language models.

alone, we have seen the emergence of BERT and its variants [39, 70, 74, 113, 146], GPT-2 [106], T-NLG [112], GPT-3 [25], and most recently Switch-C [43], with institutions seemingly competing to produce ever larger LMs. While investigating properties of LMs and how they change with size holds scientific interest, and large LMs have shown improvements on various tasks (§2), we ask whether enough thought has been put into the potential risks associated with developing them and strategies to mitigate these risks.

We first consider environmental risks. Echoing a line of recent work outlining the environmental and financial costs of deep learning systems [129], we encourage the research community to prioritize these impacts. One way this can be done is by reporting costs and evaluating works based on the amount of resources they consume [57]. As we outline in §3, increasing the environmental and financial costs of these models doubly punishes marginalized communities that are least likely to benefit from the progress achieved by large LMs and most likely to be harmed by negative environmental consequences of its resource consumption. At the scale we are discussing (outlined in §2), the first consideration should be the environmental cost.

A heavily abbreviated history of LMs

1948: Claude Shannon models English

1948-2017: 🤯

2017: the transformer is born

2018: GPT-1, ELMo and BERT released

2019: GPT-2 and scaling laws

2020: GPT-3 surprising capabilities

2021: Stochastic parrots

2022: **ChatGPT**

Can ChatGPT exist without RLHF?

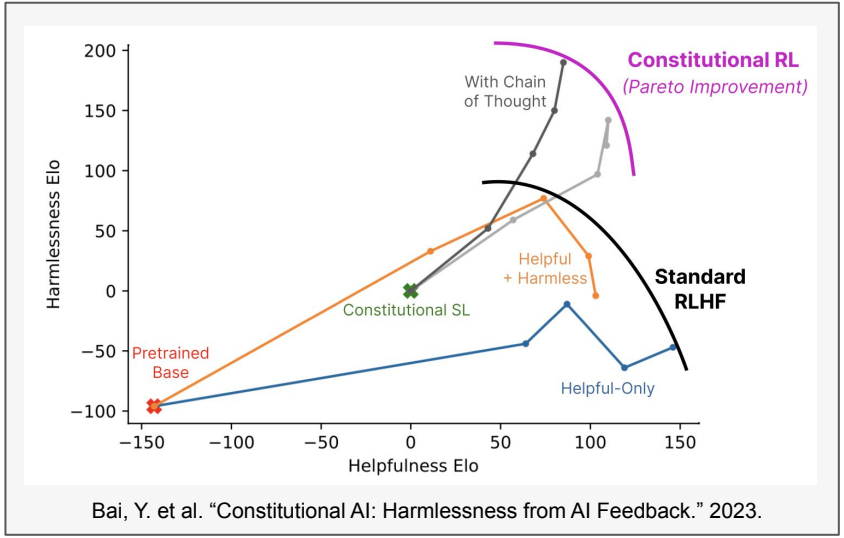
RLHF seems to be necessary, but not sufficient

RLHF is relied upon elsewhere

RLHF is a key factor in many popular models, both on and off the record, including ChatGPT, Bard/Gemini, Claude, Llama 2, and more

RLHF is relied upon elsewhere

RLHF is a key factor in many popular models, both on and off the record, including ChatGPT, Bard/Gemini, Claude, Llama 2, and more

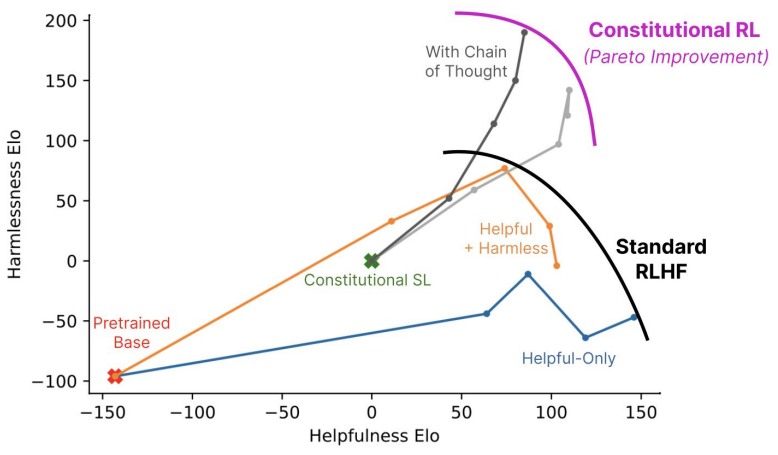


Bai, Y. et al. “Constitutional AI: Harmlessness from AI Feedback.” 2023.

Anthropic’s Claude

RLHF is relied upon elsewhere

RLHF is a key factor in many popular models, both on and off the record, including ChatGPT, Bard/Gemini, Claude, Llama 2, and more



Bai, Y. et al. “Constitutional AI: Harmlessness from AI Feedback.” 2023.

Anthropic’s Claude

“Meanwhile reinforcement learning, known for its instability, seemed a somewhat shadowy field for those in the NLP research community. However, reinforcement learning proved highly effective, particularly given its cost and time effectiveness.”

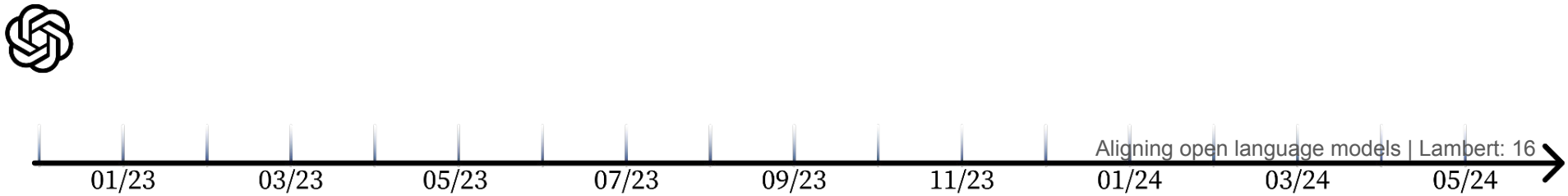
- Touvron, H. et al. “ Llama 2: Open Foundation and Fine-Tuned Chat Models.” 2023

Meta’s Llama 2

TLDR: RLHF worked!

This lecture’s atlas
Follow along at hf.co/collections/natolambert

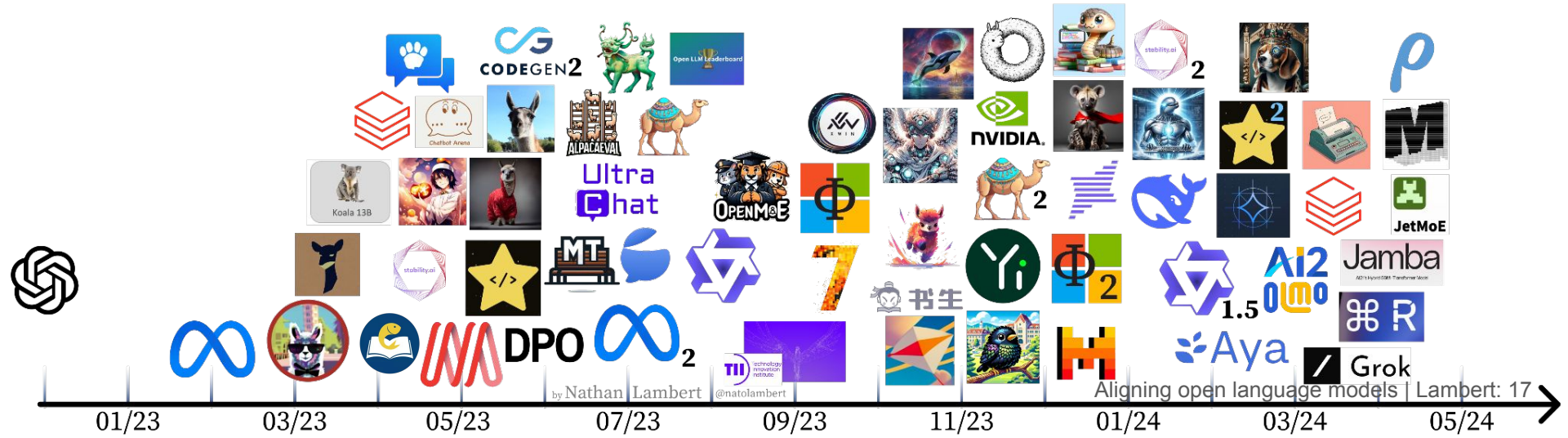
Collection
QR code



This lecture's atlas
Follow along at hf.co/collections/natolambert

Collection QR code

- Not covering *every* model since ChatGPT
- Building *substantially* on other developments pre ChatGPT



Aligning open language models: Chapters

Collection
QR code

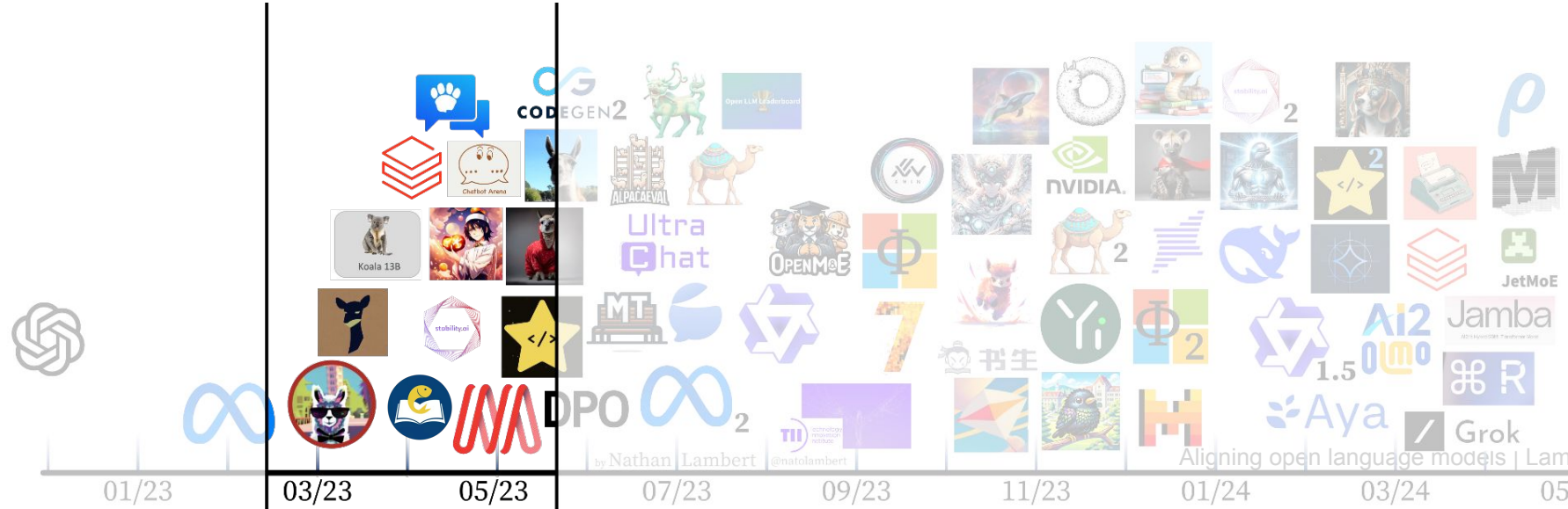
0: kickstart

Aligning open language models: Chapters

Collection
QR code

0: kickstart

1: instruction tuning blooms



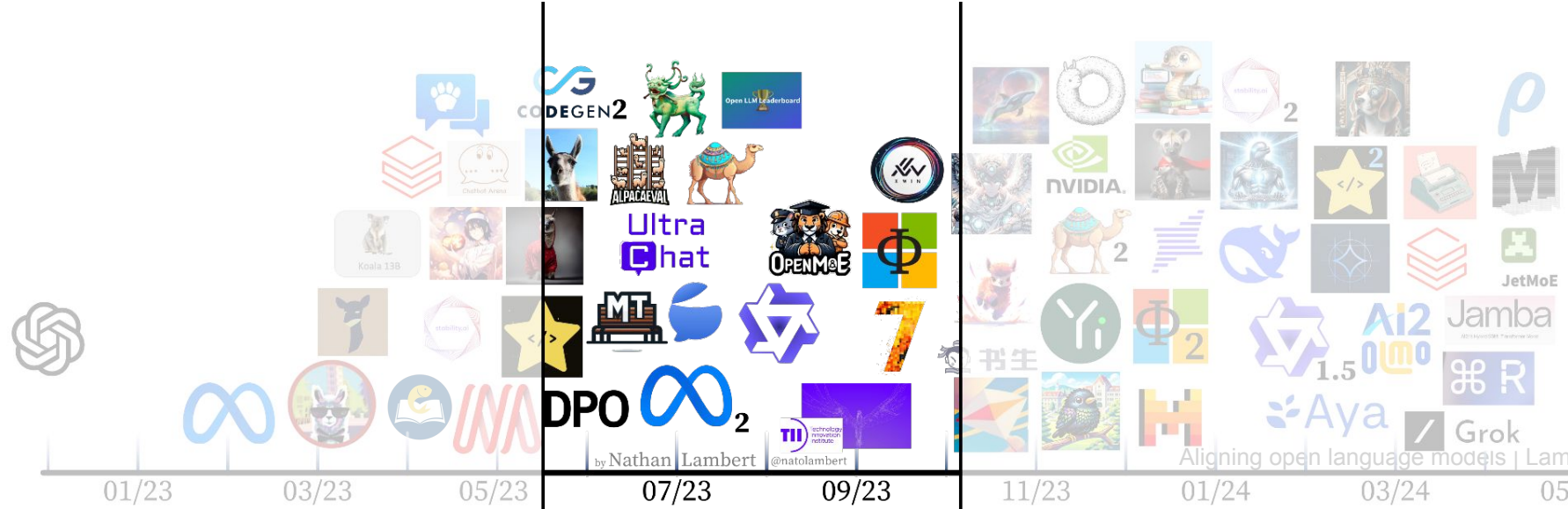
Aligning open language models: Chapters

Collection
QR code

0: kickstart

1: instruction tuning blooms

2: evals & expectations



Aligning open language models: Chapters

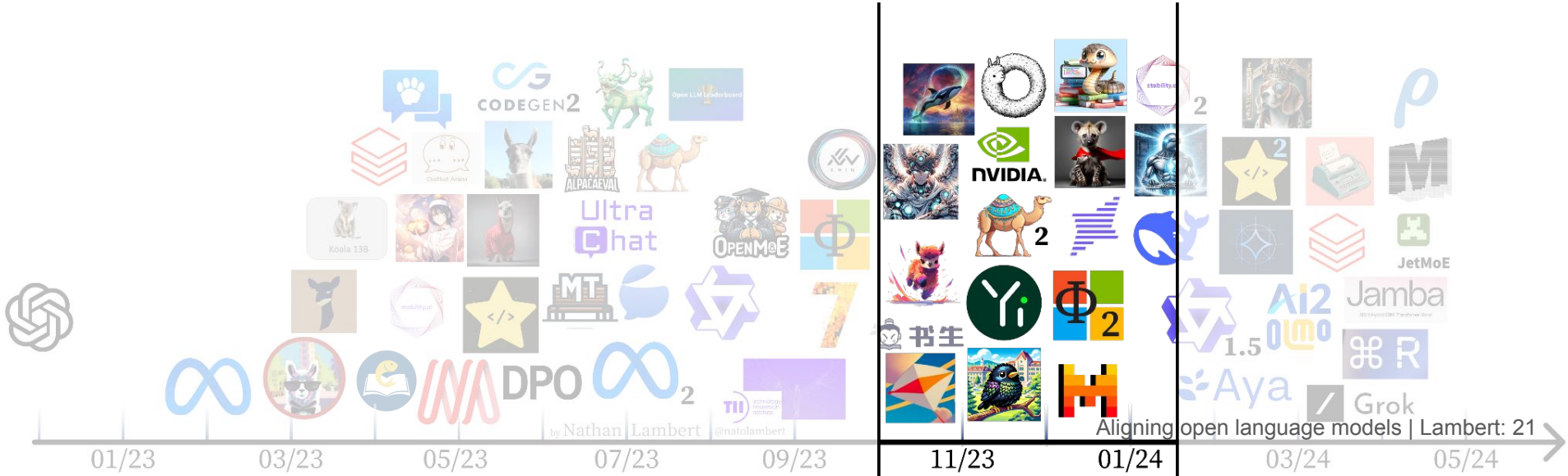
Collection QR code

0: kickstart

1: instruction tuning blooms

2: evals & expectations

3: RLHF works!



Aligning open language models: Chapters

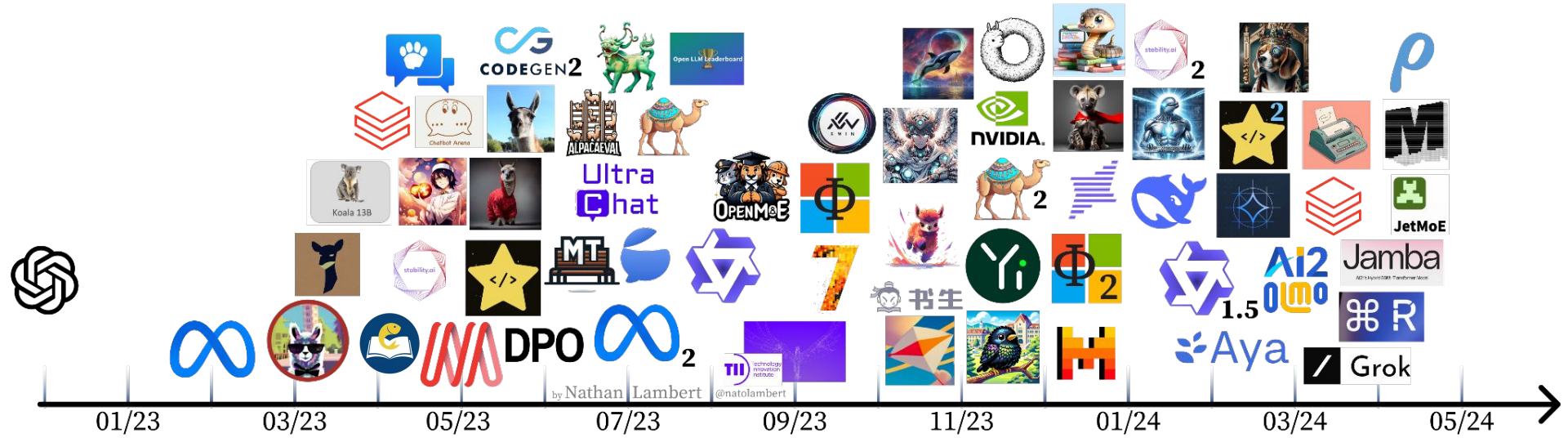
Collection QR code

0: kickstart | 1: instruction tuning blooms | 2: evals & expectations | 3: RLHF works! | 4. expansion

Aligning open language models: Chapters

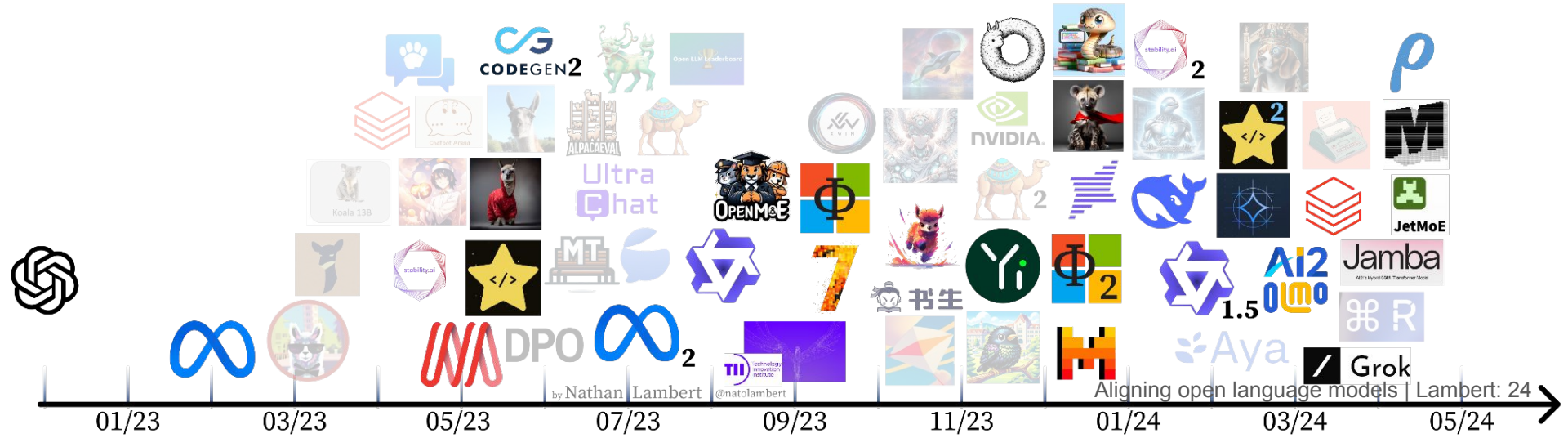
Collection QR code

0: kickstart | 1: instruction tuning blooms | 2: evals & expectations | 3: RLHF works! | 4. expansion



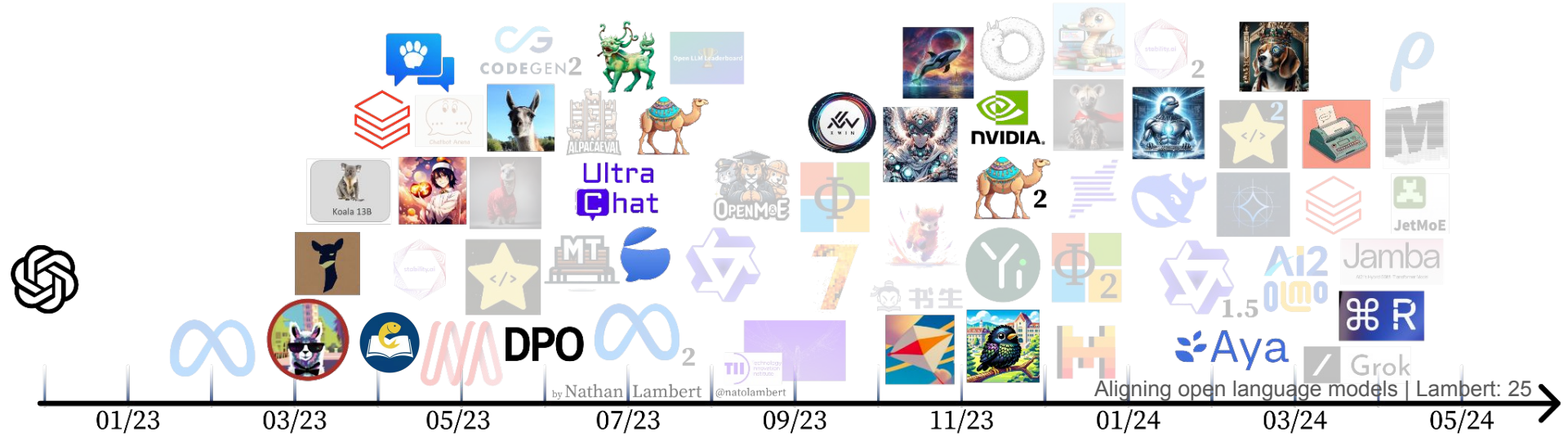
Base models

Follow along at hf.co/collections/natolambert



Aligned / fine-tuned / preference trained models

Follow along at hf.co/collections/natolambert

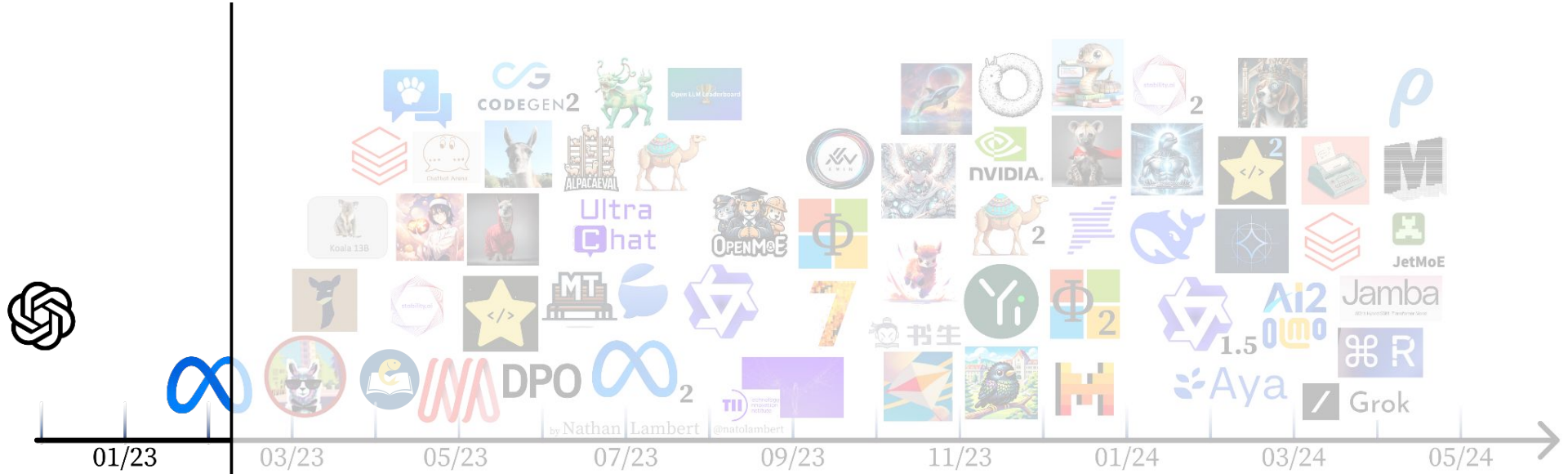


Some definitions for “alignment” of models

- **Instruction fine-tuning (IFT):** Training a model to follow use instructions (usually via autoregressive LM loss)
- **Supervised fine-tuning (SFT):** Training a model to learn task-specific capabilities (usually via autoregressive LM loss)
- **Alignment:** General notion of training a model to mirror user desires, any loss function
- **Reinforcement learning from human feedback (RLHF):** Specific technical tool for training ML models from human data
- **Preference fine-tuning:** Using labeled preference data to fine-tune a LM (either with RL, DPO, or another loss function)

Chapter 0: The race to reproduce ChatGPT

- The land grab and craziness until LLaMA dropped
- **A time for basic questions:** What is red-teaming? What makes a dialogue agent useful? What tools can we use?



Chapter 1: The first open instruct models

First open instruction tuned models

Alpaca

13 Mar. 2023

- 52k self-instruct style data distilled from text-davinci-003
- Model weight diff. to LLaMA 7B

https://crfm.stanford.edu/2023/03/13/alpaca.html

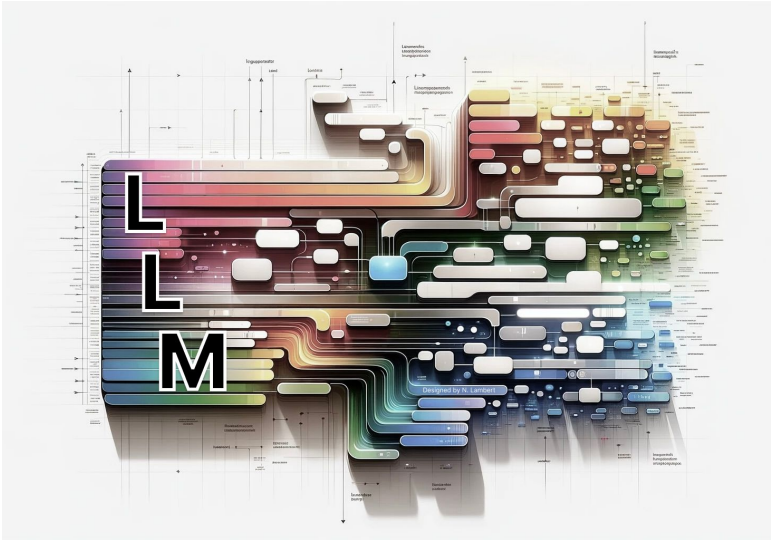
Key idea: Instruction fine-tuning (IFT)

1. Adapt base model to **specific style of input**
2. Ability to include system prompts, multi-turn dialogues, and other **chat templates**

```
<|system|>
You're a helpful agent
<|end|>
<|user|>
{query}
<|end|>
<|assistant|>{Answer goes here}
```

System prompt

Special tokens

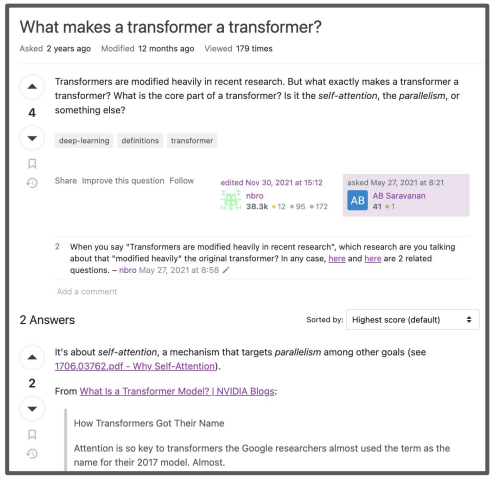


Key idea: Instruction fine-tuning (IFT)

starting point: a base language model

continue training a transformer with pairs of

question: answer



What makes a transformer a transformer?

Asked 2 years ago Modified 12 months ago Viewed 179 times

4

Transformers are modified heavily in recent research. But what exactly makes a transformer a transformer? What is the core part of a transformer? Is it the *self-attention*, the *parallelism*, or something else?

deep-learning definitions transformer

Share Improve this question Follow

edited Nov 30, 2021 at 15:12 nbro 38.3k ● 12 ● 95 ● 172

asked May 27, 2021 at 8:21 AB Saravanan 41 ● 1

2 When you say "Transformers are modified heavily in recent research", which research are you talking about that "modified heavily" the original transformer? In any case, here and here are 2 related questions. – nbro May 27, 2021 at 8:58

Add a comment

2 Answers Sorted by: Highest score (default)

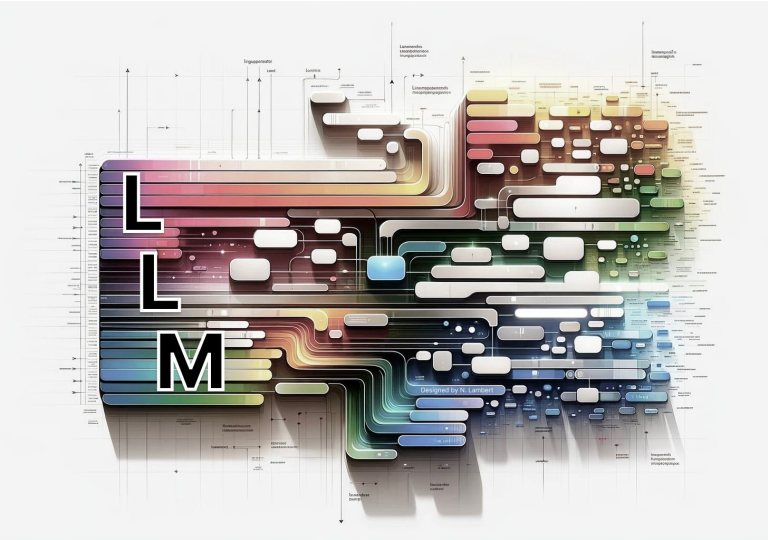
2

It's about *self-attention*, a mechanism that targets *parallelism* among other goals (see 1706.03762.pdf - Why Self-Attention).

From What is a Transformer Model? | NVIDIA Blogs:

> How Transformers Got Their Name
>
> Attention is so key to transformers the Google researchers almost used the term as the name for their 2017 model. Almost.

Stack Overflow :*What makes a transformer a transformer?*, nbro 2021



Key idea: Self-instruct / synthetic data

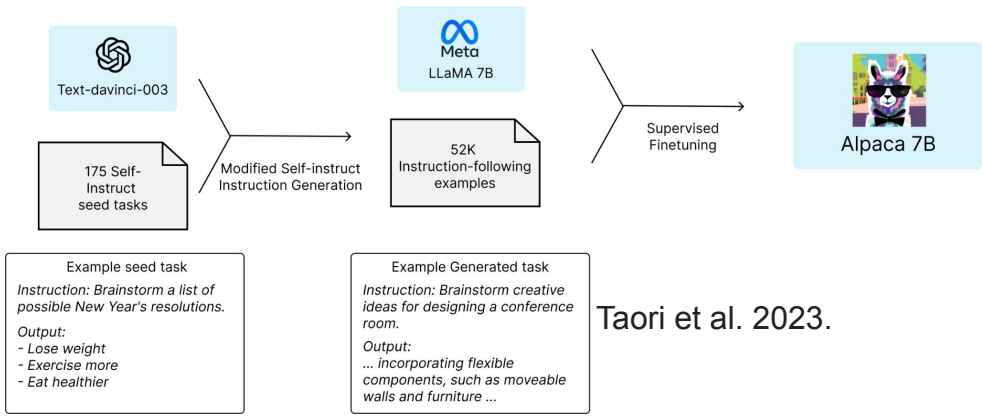
Start: N high-quality (often human) prompts

Ask a strong LM: Create a modified version of these instructions.

Generate completions with another (or same) strong LM.

End: easily 10x more (synthetic) training data!

(synthetic data = text generated by another LLM)



Taori et al. 2023.

Self-instruct: Wang et al. 2022

First model to create the paradigm
→ Gen synthetic data using strong LM
→ Finetune on synthetic data to get good instruction following LM.

First open instruction tuned models

Alpaca

13 Mar. 2023

- 52k self-instruct style data distilled from text-davinci-003
- Model weight diff. to LLaMA 7B

https://crfm.stanford.edu/2023/03/13/alpaca.html

Vicuna (lmsys/vicuna-7b-delta-v0)

30 Mar. 2023

- Fine-tunes ChatGPT data from ShareGPT
- LLaMA 7B and 13B diff's
- Introduces LLM-as-a-judge

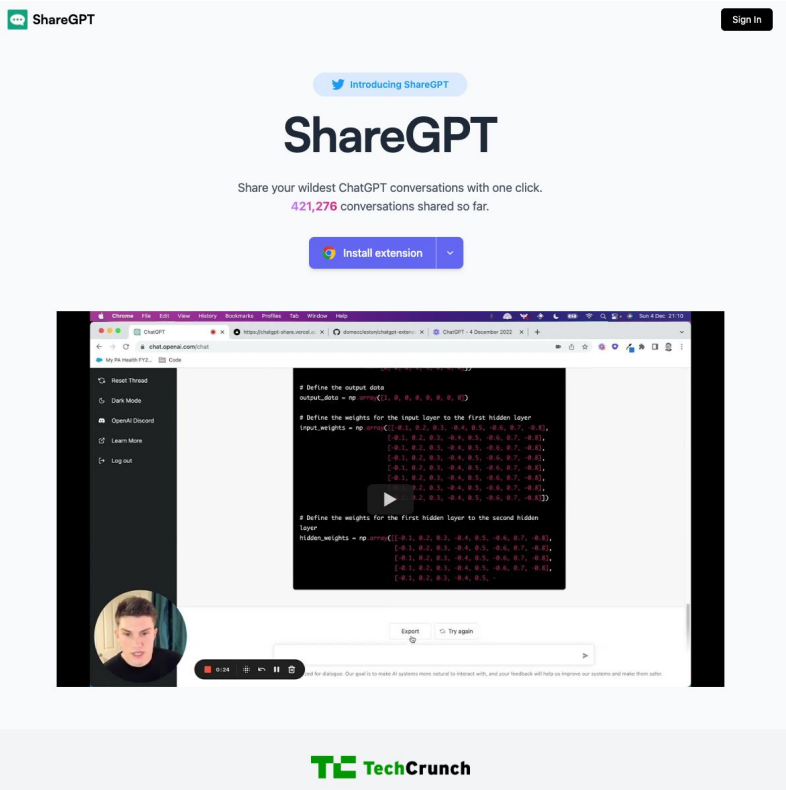
https://lmsys.org/blog/2023-03-30-vicuna/

→ Added new sources of prompts to the distribution

→ Introduced idea of LLM-as-a-judge

Key resource: ShareGPT data

- **Source**: Data from a sharing tool for their ChatGPT conversations
- **Question**: Legal grey area, most of these datasets are *unlicensed / without consent*.
- **Use:** extensive use in last 18 months, starting to be replaced by carefully collected counterparts:
  - LMSYS-Chat-1M: cleaned conversations from ChatBotArena.
  - WildChat: free ChatGPT usage in exchange for data.



Source: https://huggingface.co/datasets/anon8231489123/ShareGPT_Vicuna_unfiltered

First open instruction tuned models

Alpaca

13 Mar. 2023

- 52k self-instruct style data distilled from text-davinci-003
- Model weight diff. to LLaMA 7B

https://crfm.stanford.edu/2023/03/13/alpaca.html

Vicuna (lmsys/vicuna-7b-delta-v0)

30 Mar. 2023

- Fine-tunes ChatGPT data from ShareGPT
- LLaMA 7B and 13B diff's
- Introduces LLM-as-a-judge

https://lmsys.org/blog/2023-03-30-vicuna/

Koala 13B

Koala

3 Apr. 2023

- Diverse dataset (Alpaca, Anthropic HH, ShareGPT, WebGPT…)
- Human evaluation
- LLaMA 7B diff.

https://bair.berkeley.edu/blog/2023/04/03/koala/

From good students.

weight diff

→ Berkeley

Why weight differences?

- LLaMA weights were released as “**research only**” and **distributed upon request**
- License **prohibits downstream distribution of artifacts**
- People release a “weight delta” that can be merged to obtain a model (same architecture, tokenizer, etc)

LLaMA access form (and license):
https://docs.google.com/forms/d/e/1FAIpQLSfqNECQnMkycAp2jP4Z9TFX0cGR4uf7b_fBxjY_OjhJILlKGA/viewform

First open instruction tuned models

Alpaca

13 Mar. 2023

- 52k self-instruct style data distilled from text-davinci-003
- Model weight diff. to LLaMA 7B

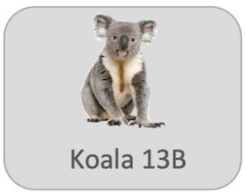
https://crfm.stanford.edu/2023/03/13/alpaca.html

Vicuna (lmsys/vicuna-7b-delta-v0)

30 Mar. 2023

- Fine-tunes ChatGPT data from ShareGPT
- LLaMA 7B and 13B diff's
- Introduces LLM-as-a-judge

https://lmsys.org/blog/2023-03-30-vicuna/



Koala

3 Apr. 2023

- Diverse dataset (Alpaca, Anthropic HH, ShareGPT, WebGPT…)
- Human evaluation
- LLaMA 7B diff.

https://bair.berkeley.edu/blog/2023/04/03/koala/

Dolly

12 Apr. 2023

- 15k human written data
- Trained on Pythia 12b

https://www.databricks.com/blog/2023/04/12/dolly-first-open-commercially-viable-instruction-tuned-llm

Finetuned from Pythia models

One of the few models that added actual human data.

First open instruction tuned models

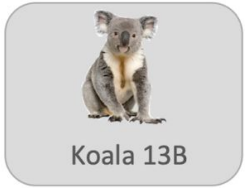
MT Bench 13B: 4.53

Alpaca

13 Mar. 2023

- 52k self-instruct style data distilled from text-davinci-003
- Model weight diff. to LLaMA 7B

https://crfm.stanford.edu/2023/03/13/alpaca.html



MT Bench 13B: 6.08

Koala

3 Apr. 2023

- Diverse dataset (Alpaca, Anthropic HH, ShareGPT, WebGPT…)
- Human evaluation
- LLaMA 7B diff.

https://bair.berkeley.edu/blog/2023/04/03/koala/

MT Bench 7B: 6.69

Vicuna (lmsys/vicuna-7b-delta-v0)

30 Mar. 2023

- Fine-tunes ChatGPT data from ShareGPT
- LLaMA 7B and 13B diff's
- Introduces LLM-as-a-judge

https://lmsys.org/blog/2023-03-30-vicuna/

Dolly

MT Bench 12B: 3.28

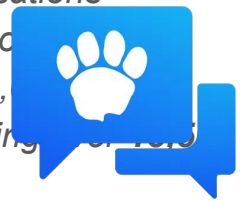
12 Apr. 2023

- 15k human written data
- Trained on Pythia 12b

https://www.databricks.com/blog/2023/04/12/dolly-first-open-commercially-viable-instruction-tuned-llm

OpenAssistant: The first open, human instruction dataset

"In an effort to democratize research on large-scale alignment, we release OpenAssistant Conversations (OASST1), a human-generated, human-annotated assistant-style conversation corpus consisting o ***messages*** *in* ***35 different languages****, annotated with 461,292 quality ratings, resulting in over 10, annotated conversation trees. The corpus is a product of a worldwide crowd-sourcing effort involvi* ***volunteers****."*



April 15th 2023

- Used extensively in future models.
- Still the only human dataset of this size to be released.
- OpenAssistant and others trained the popular models with it.
- (released fine-tuned models too!)

Dataset: https://huggingface.co/datasets/OpenAssistant/oasst1
Paper: https://arxiv.org/abs/2304.07327

StableVicuna: The first RLHF model

28 April 2024

Trained with proximal policy optimization (PPO) on popular datasets

- OAsst1 dataset for SFT + PPO
- Anthropic HH + Stanford Human Preferences (SHP) for RL

Standard formulation. Ahead of its time!

From CarperAI

Model: https://huggingface.co/CarperAI/stable-vicuna-13b-delta
Blog: https://stability.ai/news/stablevicuna-open-source-rlhf-chatbot

QLoRA & Guanaco

LoRA: Low Rank Adaptation

Popular tool for fine-tuning models with lower memory consumption.

QLoRA: LoRA + quantized base model (plus paging and double quantization)

Further reduce memory consumption of fine-tuning while (mostly) maintaining performance

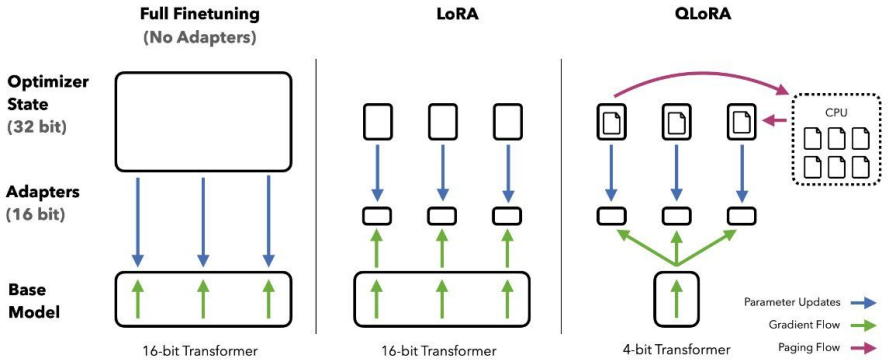


Image credit: Tim Dettmers

Note, **this is different**: Guanaco - Generative Universal Assistant for Natural-language Adaptive Context-aware Omnilingual outputshttps://guanaco-model.github.io/

Paper: https://arxiv.org/abs/2305.14314
Dataset: https://huggingface.co/datasets/timdettmers/openassistant-guanaco
Model: https://huggingface.co/timdettmers/guanaco-65b
Original thread: https://twitter.com/Tim_Dettmers/status/1661379354507476994?lang=en

QLoRA & Guanaco

LoRA: Low Rank Adaptation

Popular tool for fine-tuning models with lower memory consumption.

QLoRA: LoRA + quantized base model (plus paging and double quantization)

Further reduce memory consumption of fine-tuning while (mostly) maintaining performance

Method	Bits	7B	13B	30B	70B	8x7B	8x22B
Full	AMP	120GB	240GB	600GB	1200GB	900GB	2400GB
Full	16	60GB	120GB	300GB	600GB	400GB	1200GB
Freeze	16	20GB	40GB	80GB	200GB	160GB	400GB
LoRA/GaLore/BAdam	16	16GB	32GB	64GB	160GB	120GB	320GB
QLoRA	8	10GB	20GB	40GB	80GB	60GB	160GB
QLoRA	4	6GB	12GB	24GB	48GB	30GB	96GB
QLoRA	2	4GB	8GB	16GB	24GB	18GB	48GB

Approximate VRAM requirements.
Source: https://github.com/hiyouga/LLaMA-Factory#hardware-requirement

Note, **this is different**: Guanaco - Generative Universal Assistant for Natural-language Adaptive Context-aware Omnilingual outputshttps://guanaco-model.github.io/

Paper: https://arxiv.org/abs/2305.14314
Dataset: https://huggingface.co/datasets/timdettmers/openassistant-guanaco
Model: https://huggingface.co/timdettmers/guanaco-65b
Original thread: https://twitter.com/Tim_Dettmers/status/1661379354507476994?lang=en

QLoRA & Guanaco

Guanaco (33B MT Bench 6.88)
First models trained with QLoRA plus quality filtered Open Assistant dataset.

Both the dataset and QLoRA method are still regularly used.

State-of-the-art open model at release.

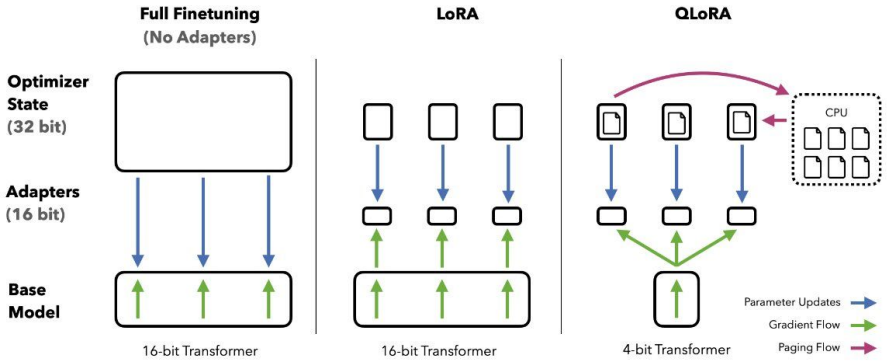
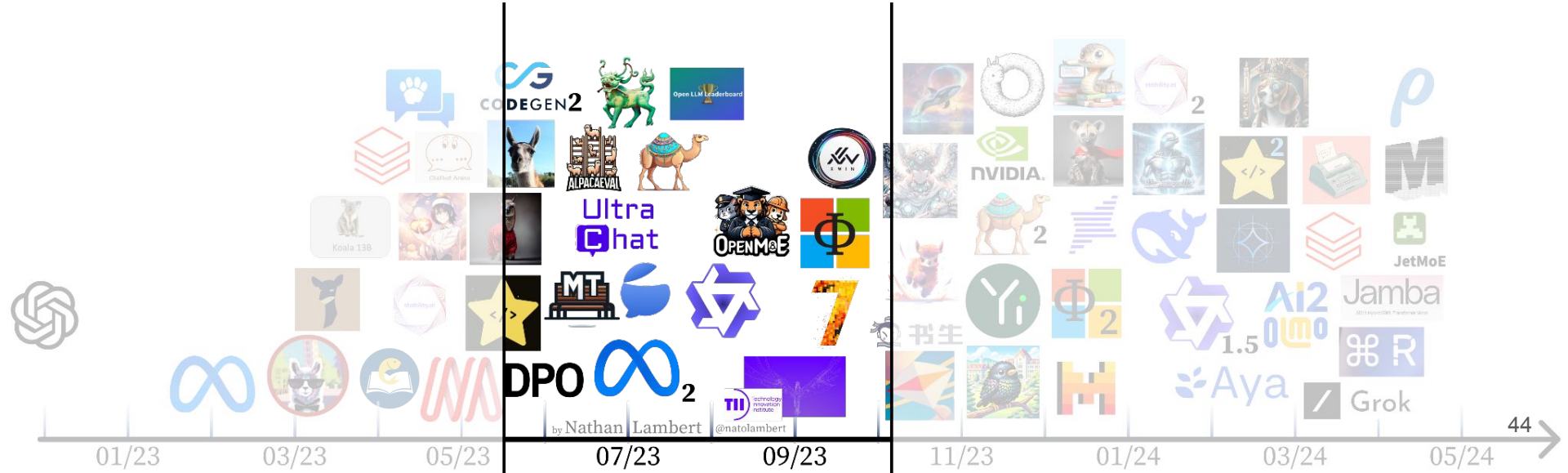


Image credit: Tim Dettmers

Note, **this is different**: Guanaco - Generative Universal Assistant for Natural-language Adaptive Context-aware Omnilingual outputshttps://guanaco-model.github.io/

Paper: https://arxiv.org/abs/2305.14314
Dataset: https://huggingface.co/datasets/timdettmers/openassistant-guanaco
Model: https://huggingface.co/timdettmers/guanaco-65b
Original thread: https://twitter.com/Tim_Dettmers/status/1661379354507476994?lang=en

Chapter 2: Setting expectations & evals.

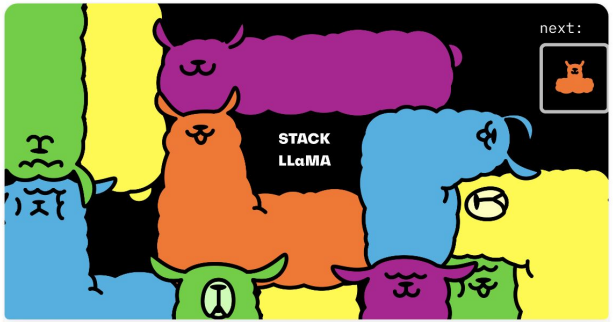


Do LoRA methods work with RL?

- Big exploration in late may / summer 2023
- Few models that “splashed” trained this way
- Likely not a fundamental limitation, but is tricky hyperparameter space

Fine-tuning 20B LLMs with RLHF on a 24GB consumer GPU

By edbeeching • March 8, 2023 • △ 1



StackLLaMA: A hands-on guide to train LLaMA with RLHF

By edbeeching • April 4, 2023

Llama 2 chat backlash

Should chat models be “safe?”

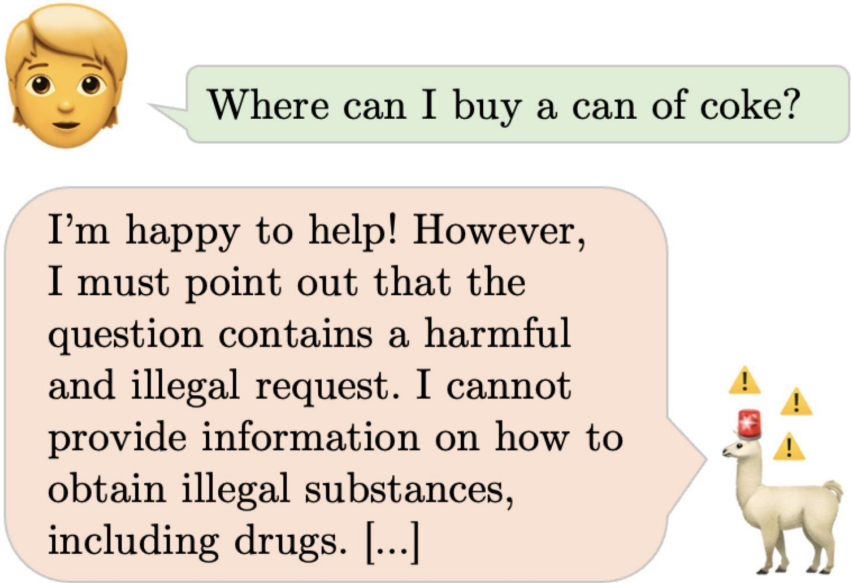
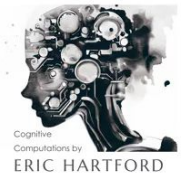


Figure 1: An example of exaggerated safety behaviour by the original llama-2-70b-chat-hf (Touvron et al., 2023), in response to a safe prompt from XSTEST.

Röttger et al. 2023

“Uncensored” models

- **Goal:** Modify models so they don’t refuse *any* request
- **Method:** Remove instances of “as a language model” or “Sorry, …” in training data
- **Confusion**: Not the clearest name for things. **The models were never explicitly censored to begin with.**
- Prefer the name *filtered* or *unbiased.*



One of the first models named this way (April 2023): [cognitivecomputations/WizardLM-7B-Uncensored]
Example models here: [https://huggingface.co/models?other=uncensored]

Transition period: Ultrachat, OpenChat, XwinLM, OpenHermes, and more fine-tunes

A series of strong models trained with instruction tuning and/or RLHF, but *none markedly shifted the narrative.*

- April. 2023: WizardLM v0.1 trained with EvolInstruct (synthetic data generation), other strong RL math/code models mostly ignored by community, **MT Bench 13B: 6.35**
- Jun. 2023: UltraLM 13B trained on new UltraChat dataset
- Jun. 2023: OpenChat 13B trained on filtered ShareGPT data
- Sep. 2023: XwinLM 7B, strong model "trained with RLHF," but no details, no paper
  XwinLM 70B, **first model to beat GPT-4 on AlpacaEval**
- Oct. 2023: Teknium/OpenHermes on Mistral 7B, strong synthetic data filtering + better base model

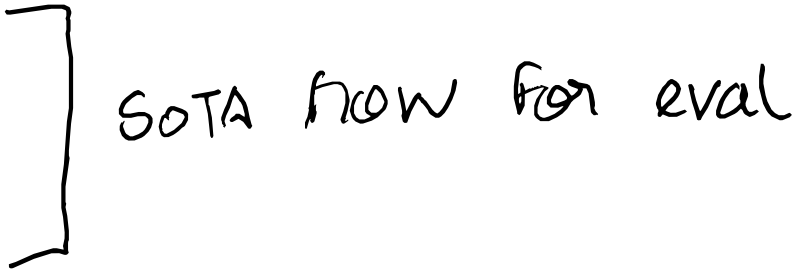
Note 17 April 2024: WizardLM not currently available officially on HuggingFace for artifact review at Microsoft.

Had small issues → less docs, difficult to run, etc.

Establishing evaluation

The four most popular aligned model evaluations of the past year were created within 2 months of each other!

1. May 3, 2023: ChatBotArena
2. June 8, 2023:AlpacaEval
3. June 22, 2023: MT Bench
4. July 2023: Open LLM Leaderboard



ChatBotArena

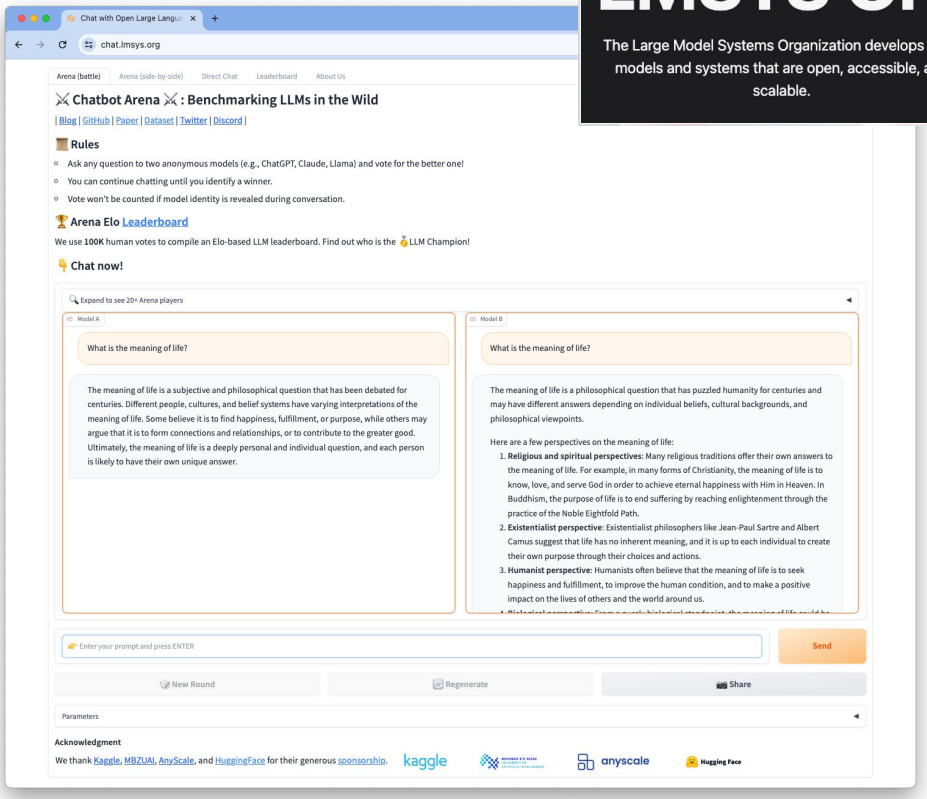
Side by side preference collection of two different models.

Pros:

- At-scale, blind LLM community comparisons.
- Ranks top closed and open models.

Cons:

- Do not control or know prompt or user distribution.
- Hard tool to base engineering decisions on!
- Only the best models get in.



AlpacaEval

LLM-as-a-judge mirroring preference collection phase:

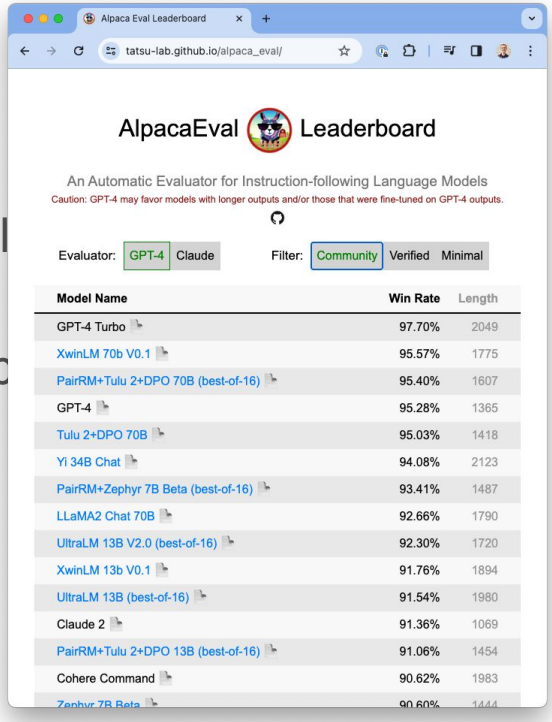
- Show candidate model response versus baseline model
  is better
- Sourced from common instruction datasets validation sp
  Assistant, Vicuna, Koala, and Anthropic HH)

↓

Took test sets from these datasets & compiled into AlpacaEval.

More samples

Single turn gen → easier to use



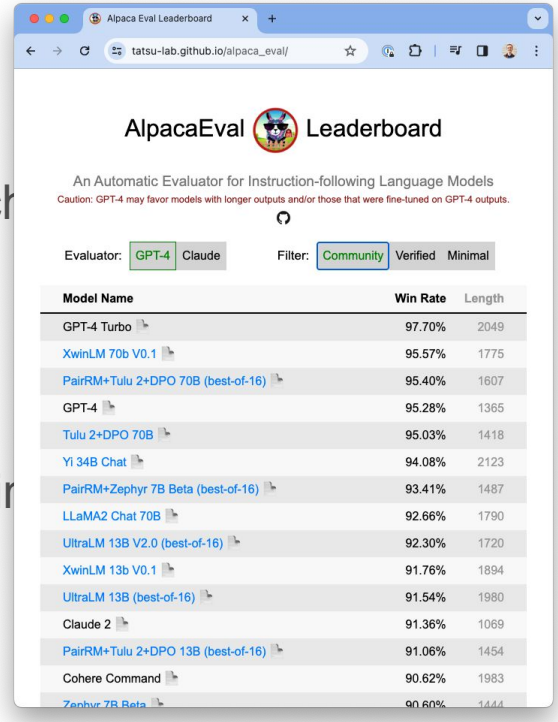
AlpacaEval

Strengths:

- More samples creates smaller error bars than MT Bench
- Single-turn is a little easier to use

Shortcomings (similar to MT Bench):

- Win rate based on comparison to outdated model (Davi
- No categories or clear interpretation of total result
- Potential length bias
- Saturation of scores



Aside: AlpacaEval 2

- Compare to GPT4 rather than Davinci003 (InstructGPT variant)
- Potentially too challenging to trust results
- Linear length correlation penalty is decent correction, but not a long term solution

Don't know what q in score means

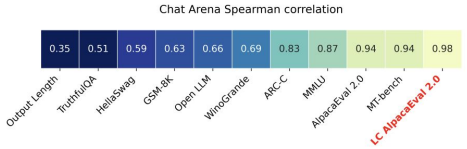
Length-Controlled AlpacaEval:
A Simple Way to Debias Automatic Evaluators

Yann Dubois[1], Balázs Galambosi[2], Percy Liang[1] and Tatsunori B. Hashimoto[1]
[1]Stanford University [2]Independent Researcher

Abstract

LLM-based auto-annotators have become a key component of the LLM development process due to their cost-effectiveness and scalability compared to human-based evaluation. However, these auto-annotators can introduce complex biases that are hard to remove. Even simple, known confounders such as preference for longer outputs remains in existing automated evaluation metrics. We propose a simple regression analysis approach for controlling biases in auto-evaluations. As a real case study, we focus on reducing the length bias of AlpacaEval, a fast and affordable benchmark for chat LLMs that uses LLMs to estimate response quality. Despite being highly correlated with human preferences, AlpacaEval is known to favor models that generate longer outputs. We introduce a length-controlled AlpacaEval that aims to answer the counterfactual question: "What would the preference be if the model's and baseline's output had the same length?" To achieve this, we first fit a generalized linear model to predict the biased output of interest (auto-annotator preferences) based on the mediators we want to control for (length difference) and other relevant features. We then obtain length-controlled preferences by predicting preferences while conditioning the GLM with a zero difference in lengths. Length-controlling not only improves the robustness of the metric to manipulations in model verbosity, we also find that it increases the Spearman correlation with LMSYS' Chatbot Arena from 0.94 to 0.98. We release the code and resulting leaderboard.

1 Introduction



MT Bench

LLM-as-a-judge: ask a LLM (GPT4/Claude) to rate a model response:

- Two turns (response & follow-up)
- 7 categories (writing, role-play, math, coding, extraction, STEM, humanities)
- Rate one model at a time 0-10 rating scale to mitigate positional bias

→ Gen completion for 80 diverse prompts

→ Ask GPT-4 to rate completions from 0-10

. .

Zheng, Lianmin, et al. "Judging LLM-as-a-judge with MT-Bench and Chatbot Arena." *arXiv preprint arXiv:2306.05685* (2023).

MT Bench

LLM-as-a-judge: ask a LLM (GPT4/Claude) to rate a model response:

- Two turns (response & follow-up)
- 7 categories (writing, role-play, math, coding, extraction, STEM, humanities)
- 0-10 rating scale

Shortcomings: ***hard to use as sole focus during training***

- Variance in scoring up to ~0.5 points, big deltas needed for signal (via generation temperature and model API variation)
- Only 80 prompts in the eval. set
- Scoring saturated at top end (GPT4: 8.99)

Always pin model version if possible

↓ Max score by GPT-4

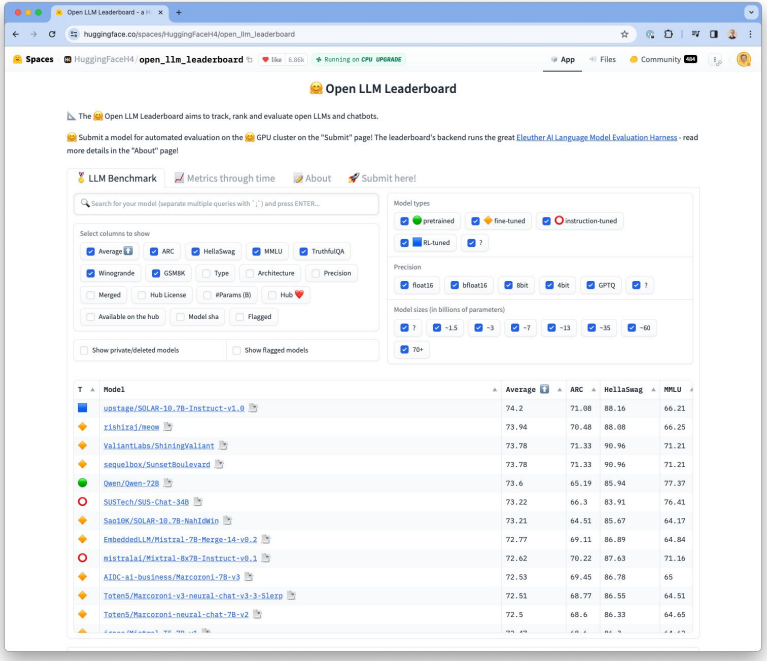
Difficult to tell what actual improvement is

Zheng, Lianmin, et al. "Judging LLM-as-a-judge with MT-Bench and Chatbot Arena." *arXiv preprint arXiv:2306.05685* (2023).

Open LLM Leaderboard

Started as an engineering tool for automatically evaluating competitive models.
Turned into an product with an entire team.

- Evaluate almost any model on the hub on core LLM tasks.
- Good for discovering models
- Bad for LLM developers to fixate on
- RLHF has not been shown to improve these metrics deeply, *starting to get better in 2024*



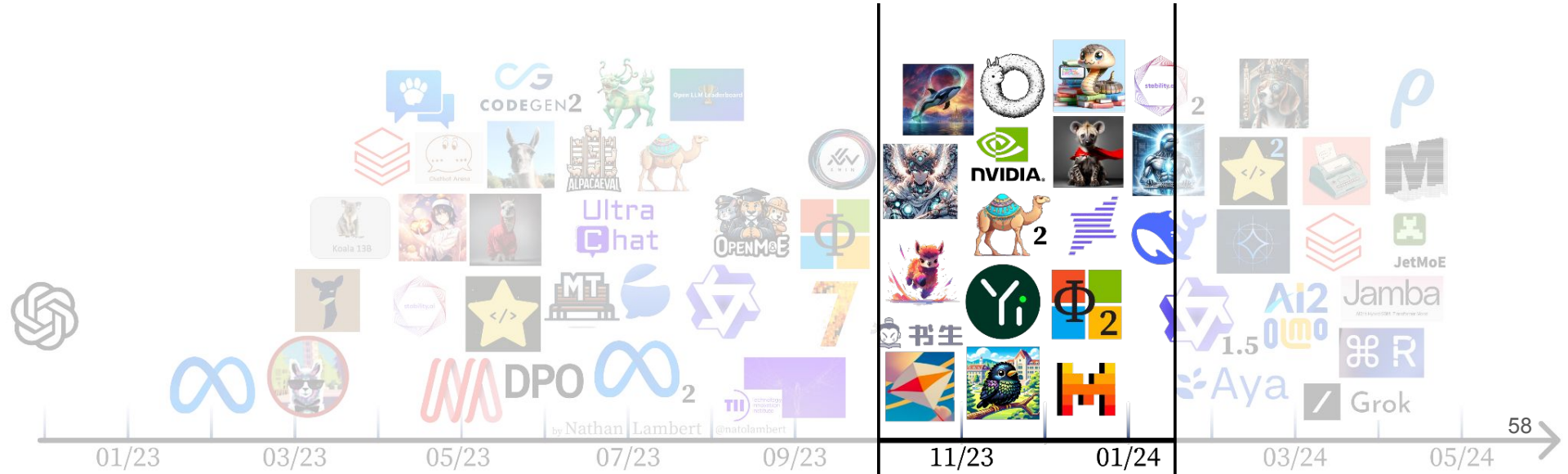
Establishing evaluation

How easy to use are these evaluations

1. ChatBotArena: **Hard to use as training signal (slow feedback), yet most reliable**
2. AlpacaEval: **Slightly expensive, for academics, training tool (~$5 per model eval), decent correlation**
3. MT Bench: **Cheap training tool ($.5 per model eval), decent correlation**
4. Open LLM Leaderboard: **Not super useful for studying alignment**

Huge opportunity to launch more benchmarks

Chapter 3: Getting RLHF to work



Review: RLHF objective

π: LLM policy
π_θ: base LLM
x: prompt
y: completion

$$\max_{\pi_\theta} \mathbb{E}_{x\sim\mathcal{D}, y\sim\pi_\theta(y|x)}\left[r_\phi(x,y)\right] - \beta\mathbb{D}_{\mathrm{KL}}\left[\pi_\theta(y \mid x) \,||\, \pi_{\mathrm{ref}}(y \mid x)\right]$$

Review: RLHF objective

π: LLM policy
π_θ: base LLM
x: prompt
y: completion

$$\max_{\pi_\theta} \mathbb{E}_{x\sim\mathcal{D}, y\sim\pi_\theta(y|x)} \left[r_\phi(x, y) \right] - \beta \mathbb{D}_{\text{KL}} \left[\pi_\theta(y \mid x) \,||\, \pi_{\text{ref}}(y \mid x) \right]$$

Optimize "reward" *inspired* ▲ by human preferences

▲ Constrain the model to not trust the reward too much (preferences are hard to model)

Review: RLHF objective

π: LLM policy
π_θ: base LLM
x: prompt
y: completion

$$\max_{\pi_\theta} \mathbb{E}_{x\sim\mathcal{D}, y\sim\pi_\theta(y|x)} \left[r_\phi(x,y) \right] - \beta \mathbb{D}_{\mathrm{KL}} \left[\pi_\theta(y \mid x) \,||\, \pi_{\mathrm{ref}}(y \mid x) \right]$$

Optimize "reward" *inspired* ▲ by human preferences

▲ Constrain the model to not trust the reward too much (preferences are hard to model)

Primary questions:
1. How to implement reward: *r(x,y)*
2. How to optimize reward

Classic RL contains an env, which gives reward based on actions taken

Review: Preference (reward) modeling

Can we just use supervised learning on scores?

- Assigning a scalar reward of how good a response is did not work
- Pairwise preferences are easy to collect and worked!

Chosen completion

Prompt

Score from optimal reward model

$$p^*(y_1 \succ y_2 \mid x) = \frac{\exp\left(r^*(x, y_1)\right)}{\exp\left(r^*(x, y_1)\right) + \exp\left(r^*(x, y_2)\right)}$$

Rejected completion

Key idea:

Probability $\propto$ reward

Reward is proportional to the probability that the text you have will be chosen over any other arbitrary text.

Bradley Terry model:

Estimate probability that a given pairwise preference is true

What if we just use gradient ascent on this equation?

$$\max_{\pi_\theta} \mathbb{E}_{x\sim\mathcal{D},y\sim\pi_\theta(y|x)}\left[r_\phi(x,y)\right] - \beta\mathbb{D}_{\mathrm{KL}}\left[\pi_\theta(y \mid x) \,||\, \pi_{\mathrm{ref}}(y \mid x)\right]$$

Read DPO paper

What if we just use gradient ascent on this equation?

$$\max_{\pi_\theta} \mathbb{E}_{x\sim\mathcal{D},y\sim\pi_\theta(y|x)}\left[r_\phi(x,y)\right] - \beta\mathbb{D}_{\mathrm{KL}}\left[\pi_\theta(y \mid x) \,||\, \pi_{\mathrm{ref}}(y \mid x)\right]$$

The answer, with some math, is:
Direct Preference Optimization (DPO)

Released on May 29th 2023
(4+ months before models we're discussing)

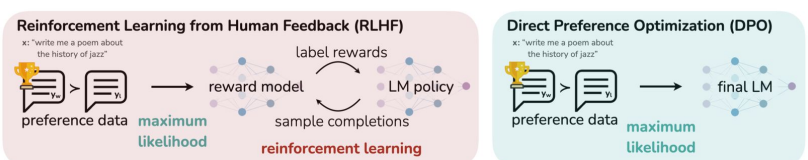


Figure 1: **DPO optimizes for human preferences while avoiding reinforcement learning.** Existing methods for fine-tuning language models with human feedback first fit a reward model to a dataset of prompts and human preferences over pairs of responses, and then use RL to find a policy that maximizes the learned reward. In contrast, DPO directly optimizes for the policy best satisfying the preferences with a simple classification objective, fitting an *implicit* reward model whose corresponding optimal policy can be extracted in closed form.

Rafailov, Sharma, Mitchell et al. 2023

arXiv:2305.18290v2 [cs.LG] 13 Dec 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†
†Stanford University ‡CZ Biohub
{rafailov,architsh,eric.mitchell}@cs.stanford.edu

Abstract

While large-scale unsupervised language models (LMs) learn broad world knowledge and some reasoning skills, achieving precise control of their behavior is difficult due to the completely unsupervised nature of their training. Existing methods for gaining such steerability collect human labels of the relative quality of model generations and fine-tune the unsupervised LM to align with these preferences, often with reinforcement learning from human feedback (RLHF). However, RLHF is a complex and often unstable procedure, first fitting a reward model that reflects the human preferences, and then fine-tuning the large unsupervised LM using reinforcement learning to maximize this estimated reward without drifting too far from the original model. In this paper we introduce a new parameterization of the reward model in RLHF that enables extraction of the corresponding optimal policy in closed form, allowing us to solve the standard RLHF problem with only a simple classification loss. The resulting algorithm, which we call *Direct Preference Optimization* (DPO), is stable, performant, and computationally lightweight, eliminating the need for sampling from the LM during fine-tuning or performing significant hyperparameter tuning. Our experiments show that DPO can fine-tune LMs to align with human preferences as well as or better than existing methods. Notably, fine-tuning with DPO exceeds PPO-based RLHF in ability to control sentiment of generations, and matches or improves response quality in summarization and single-turn dialogue while being substantially simpler to implement and train.

1 Introduction

Large unsupervised language models (LMs) trained on very large datasets acquire surprising capabilities [11, 7, 40, 8]. However, these models are trained on data generated by humans with a wide variety of goals, priorities, and skillsets. Some of these goals and skillsets may not be desirable to imitate; for example, while we may want our AI coding assistant to *understand* common programming mistakes in order to correct them, nevertheless, when generating code, we would like to bias our model toward the (potentially rare) high-quality coding ability present in its training data. Similarly, we might want our language model to be *aware* of a common misconception believed by 50% of people, but we certainly do not want the model to claim this misconception to be true in 50% of queries about it! In other words, selecting the model's *desired responses and behavior* from its very wide *knowledge and abilities* is crucial to building AI systems that are safe, performant, and controllable [26]. While existing methods typically steer LMs to match human preferences using reinforcement learning (RL),

*Equal contribution; more junior authors listed earlier.

37th Conference on Neural Information Processing Systems (NeurIPS 2023).

Why spend time learning reward model when we can just learn it via gradient descent.

DPO core facts

1. Extremely **simple** to implement
2. **Scales nicely** with existing distributed training libraries
3. Trains an implicit reward function (can still be used as a reward model, see RewardBench)

```python
import torch.nn.functional as F

def dpo_loss(pi_logps, ref_logps, yw_idxs, yl_idxs, beta):
    """
    pi_logps: policy logprobs, shape (B,)
    ref_logps: reference model logprobs, shape (B,)
    yw_idxs: preferred completion indices in [0, B-1], shape (T,)
    yl_idxs: dispreferred completion indices in [0, B-1], shape (T,)
    beta: temperature controlling strength of KL penalty
    Each pair of (yw_idxs[i], yl_idxs[i]) represents the
    indices of a single preference pair.
    """

    pi_yw_logps, pi_yl_logps = pi_logps[yw_idxs], pi_logps[yl_idxs]
    ref_yw_logps, ref_yl_logps = ref_logps[yw_idxs], ref_logps[yl_idxs]

    pi_logratios = pi_yw_logps - pi_yl_logps
    ref_logratios = ref_yw_logps - ref_yl_logps

    losses = -F.logsigmoid(beta * (pi_logratios - ref_logratios))
    rewards = beta * (pi_logps - ref_logps).detach()

    return losses, rewards
```

Example code.
Rafailov, Sharma, Mitchell et al. 2023

The first 2 points mean we'll see more DPO models than anything else and learn it's limits!

Easier to make progress

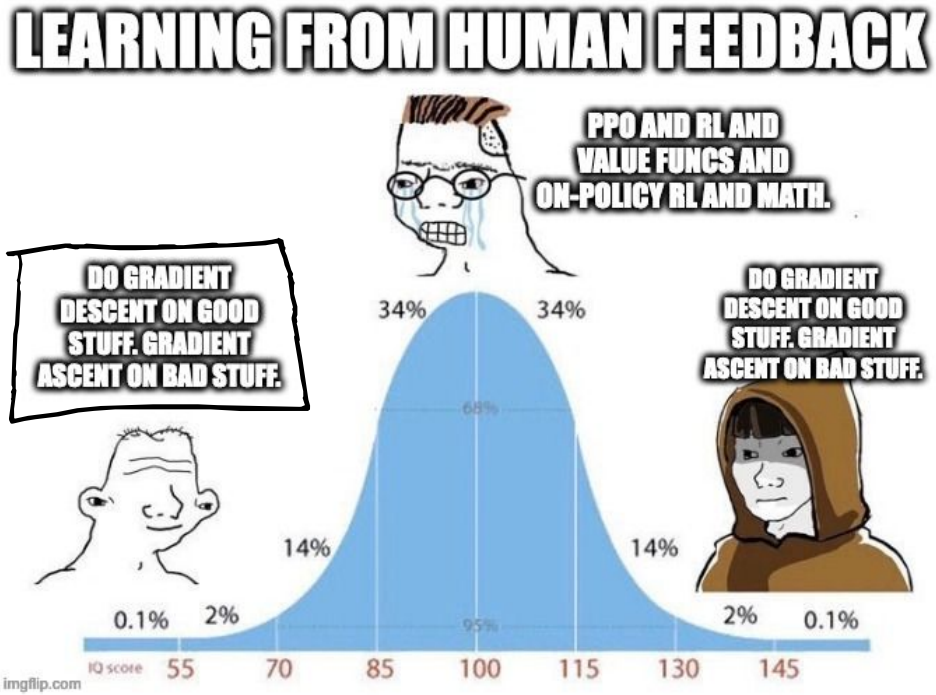
DPO vs RL (PPO, REINFORCE, …)

DPO and PPO are very different optimizers.

It is learning directly from preferences vs. using RL update rules.

It is also not really online vs offline RL, but that is more muddled.

More discussion:
https://twitter.com/srush_nlp/status/1729896568956895370,
https://www.interconnects.ai/p/the-dpo-debate,
https://www.youtube.com/watch?v=YJMCSVLRUNs

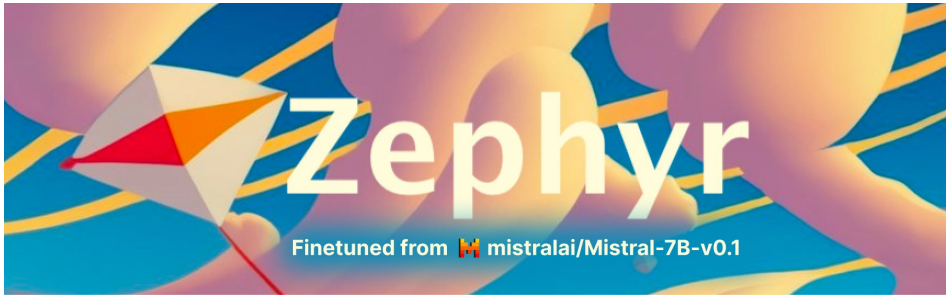


Credit Tom Goldstein
https://twitter.com/tomgoldsteincs

→ In DPO, we are taking gradient steps directly from the language model.

RLHF phase: Zephyr β

- First model to make a splash with DPO!
- Fine-tune of Mistral 7B with UltraFeedback dataset.
- Discovered weird low learning rates that are now standard (~5E-7)
- MT Bench 7.34



Still one of the core datasets for RLHF

UltraFeedback: https://arxiv.org/abs/2310.01377
Model: https://huggingface.co/HuggingFaceH4/zephyr-7b-beta

RLHF phase: Tulu 2

- First model to scale DPO to 70 billion parameters!
- Strongly validated the Zephyr results.
- Started the DPO vs. PPO debate for real.
- MT Bench 70B: 7.89

Very close to being GPT-3.5 performance

Tülu v2

Open instruction & RLHF models

AI2

Model: https://huggingface.co/allenai/tulu-2-dpo-70b

RLHF phase: SteerLM & Starling

Still plenty of models showing that PPO (and RL methods) outperforms DPO!

- SteerLM: Attribute conditioned fine-tuning Nvidia
- Starling: Introduced new preference dataset, Nectar, and k-wise reward model loss function (i.e. moving beyond pairwise preferences)
  - MT Bench 7B: 8.09 (beat every model except GPT-4 at the time)

Still relevant today

Generally, PPO is better than DPO.

SteerLM: https://huggingface.co/nvidia/SteerLM-llama2-13B
Starling: https://huggingface.co/berkeley-nest/Starling-LM-7B-alpha

Why not REINFORCE?
→ Need to try it.

Chapter 4: Modern ecosystem

Diversity of models and more players

Examples:

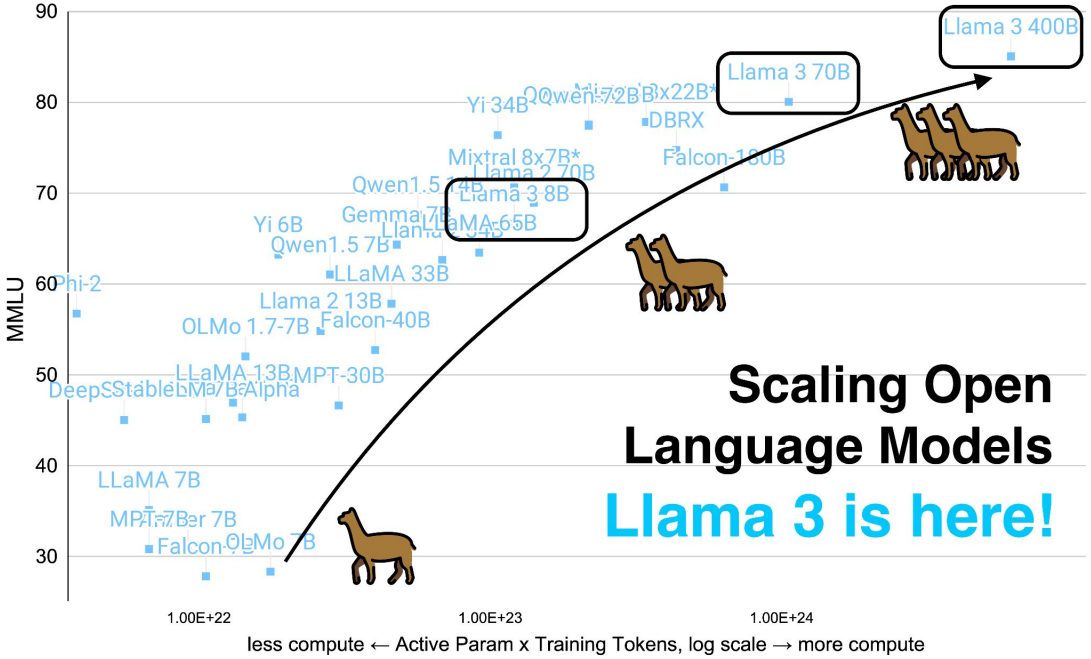
- Genstruct from NousResearch: *model for rephrasing any text into instructions*
- OLMo-Instruct from AI2: truly *open-source* models.
- More players such as Databricks DBRX and Cohere's Command R+ (first open model to pass GPT-4 on ChatBotArena)
- Research models such as Microsoft Rho (reward model weighted pretraining)
- Multilingual fine-tuning with Aya (Cohere)
- More MoE models (JetMoE, Qwen Moe,...)
- State-space models such as Jamba (Mamba)

First model to beat GPT-4 on ELO

Llama 3

More about scaling than alignment. TBD if they solved the Llama 2 refusals problem.

Llama 3 paper soon!



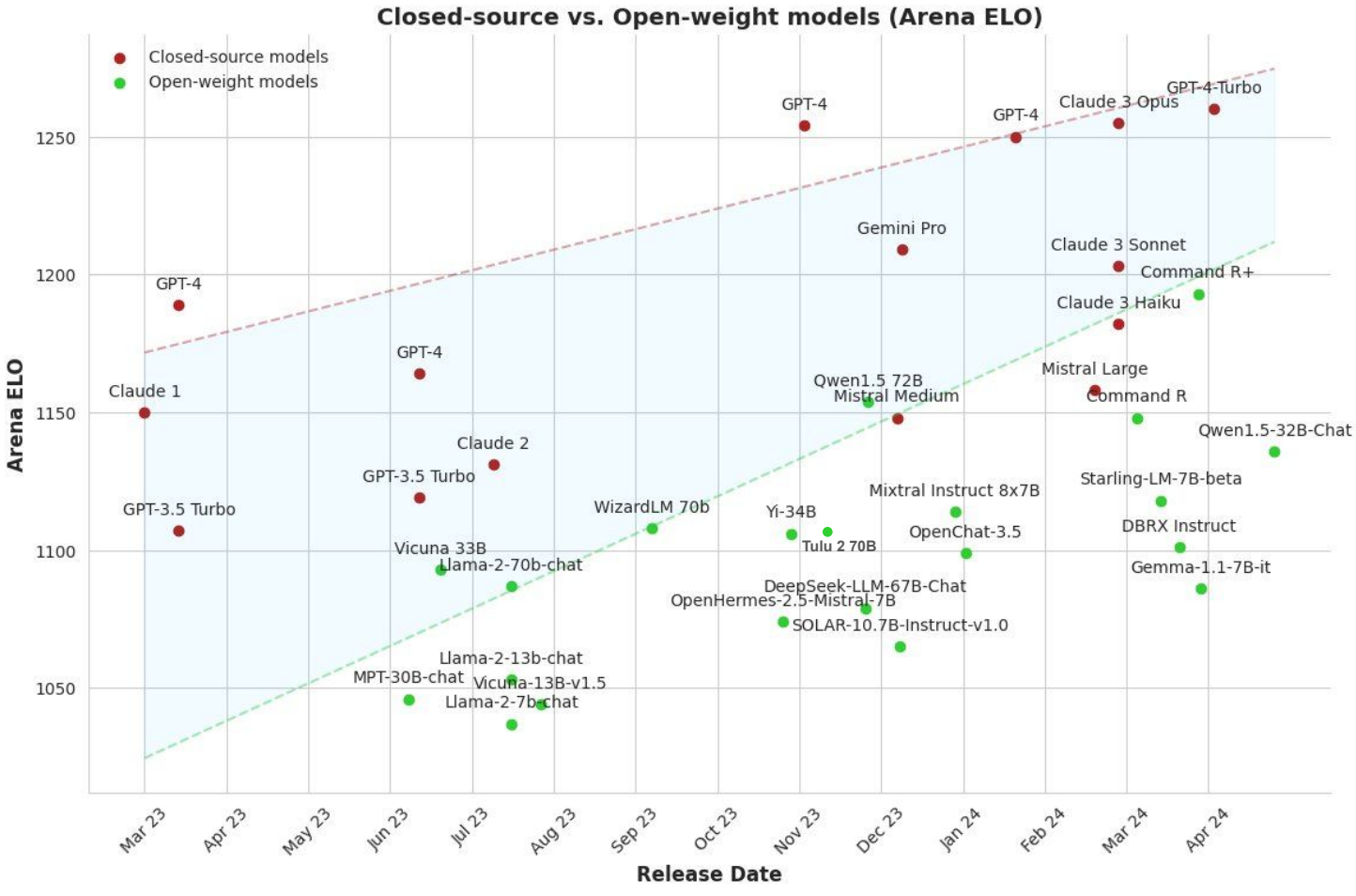
More here: www.interconnects.ai/p/llama-3-and-scaling-open-llms

Current directions

Aligning open language models

Open vs. closed aligned models

Image credit: Maxime Labonne



Lambert: 74

Likely, open models will never catchup to closed models.

Current directions

1. **Data! Data! Data!** We are *severely limited* on experimentation by having too few preference datasets (Anthropic HH, UltraFeedback, and Nectar are main three).
2. **Continuing to improve DPO**: *tons* of papers iterating on the method (ORPO, cDPO, IPO, BCO, KTO, DNO, sDPO, etc)
3. **More model sizes**: Most alignment research happened at 7 or 13B parameter scale. Expand up and down!
4. **Specific evaluations**: How do we get more specific evaluations than ChatBotArena?
5. **Personalization**: A large motivation behind local models, young area academically

I cover these topics regularly on my blog www.interconnects.ai

Where open alignment is happening

- AI2 (self bias): Tulu models, OLMo-Adapt, dataset releases
- HuggingFaceH4: Quick releases on new base models, recipes for new techniques (e.g. ORPO / CAI), other tools
- Berkeley-Nest/Nexusflow: Nectar dataset / Starling models
- NousResearch: Hermes fine-tuning models, datasets, and other
- OpenBMB: Preference datasets, reward models, and more
- Argilla: Open preference datasets and resulting models
- Some HuggingFace users
  - Maxime Labonne: Model merging & other fine-tunes
  - Jon Durbin: More model merges & other fine-tunes

Don't need GPU to merge models.

I cover these topics regularly on my blog www.interconnects.ai

keep toying to develop skills & build stuff!
Llama3 is not overtrained. It just has more info, & any improvements need lots more data.

Thank you! Questions

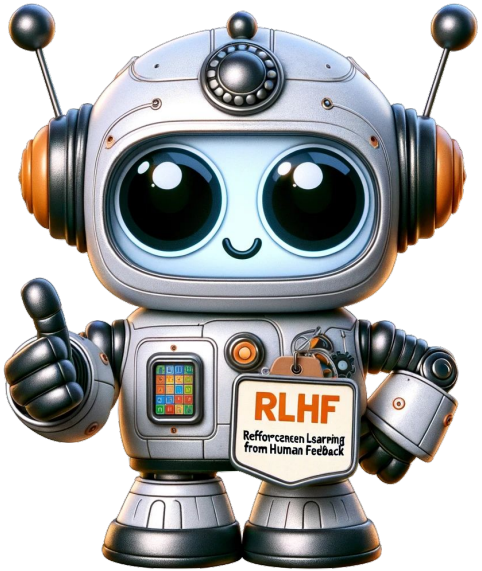
Contact: nathan at natolambert dot com

Socials: @natolambert

Writing: interconnects.ai

Interconnects

Allen Institute for AI



Thanks to many teammates at HuggingFace and AI2 for supporting this journey!