Large Language Models and Factuality

Part 1: Modern LLMs

AthensNLP

Athens, September 24 2024

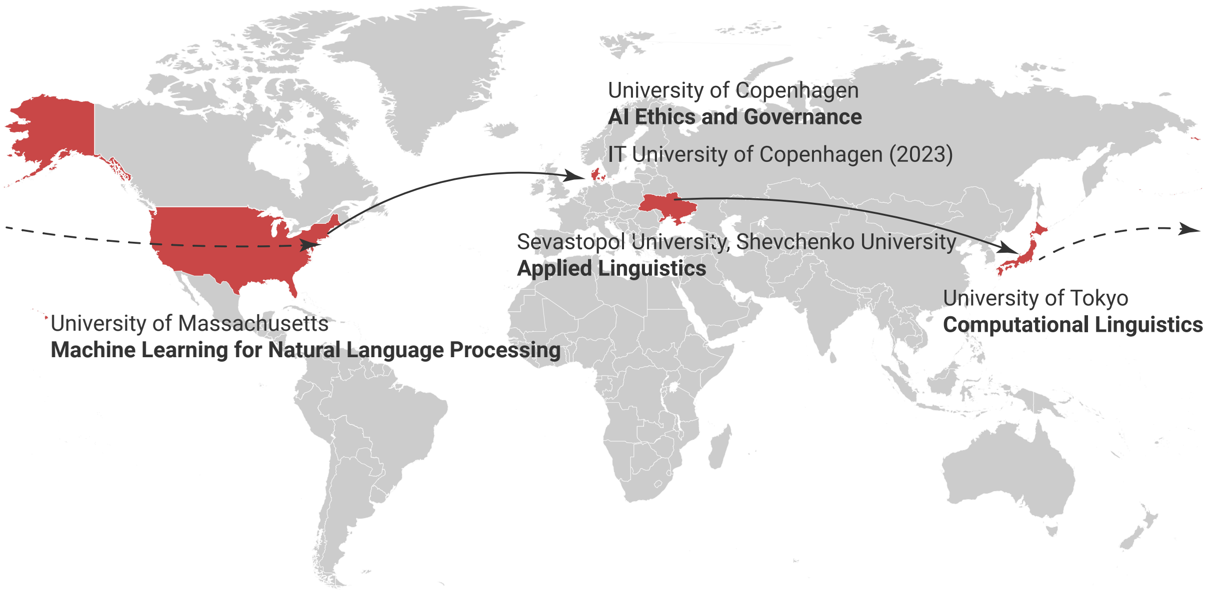
Anna Rogers

📢 🇩🇰 **open PhD and postdoc positions!**

IT UNIVERSITY OF COPENHAGEN

Anna Rogers (Assoc. Prof. @ ITU Copenhagen 🇩🇰)

- Main research areas: analysis and evaluation of Large Language Models (LLMs), AI and society
- Also: meta-science, peer review (program chair at ACL'23, co-editor-in-chief of ARR 2024-2025, led the first ChatGPT policy development)



Before we start: what's your current take?

In this lecture:

1. Modern LLMs
2. Facts *from* LLMs
3. Facts *on* LLMs

Part 1. Modern LLMs

- Modern LLMs: what do we even mean?
- In-weights vs in-context learning
- Instruction tuning
- Optimizing for preferences

MODERN LLM-BASED SYSTEMS

What counts as an LLM?

- models ~~language~~ text
- trained on at least 1B tokens
- is used for transfer learning

CF: 'foundation model', 'frontier model'

Rogers, Luccioni (2024) Position: Key Claims in LLM Research Have a Long Tail of Footnotes

LMs are actually *corpus* models

we would... propose a change from the theory-laden term ***language model*** *to the more objectively accurate term* ***corpus model****. Not only does the term corpus model better reflect the contents of models, it also provides transparency in discussing issues such as model bias. One might be surprised if a language model is biased, or if there is different bias in two different language models, but a bias in corpus models and different biases in different corpus models is almost an expectation. Natural language is not biased. What people say or write can be biased*

Veres (2022) Large Language Models are Not Models of Natural Language: They are Corpus Models

It's not just the linguists saying that!

Andrej Karpathy
@karpathy

It's a bit sad and confusing that LLMs ("Large Language Models") have little to do with language; It's just historical. They are highly general purpose technology for statistical modeling of token streams. A better name would be Autoregressive Transformers or something.

They don't care if the tokens happen to represent little text chunks. It could just as well be little image patches, audio chunks, action choices, molecules, or whatever. If you can reduce your problem to that of modeling token streams (for any arbitrary vocabulary of some set of discrete tokens), you can "throw an LLM at it".

https://x.com/karpathy/status/1835024197506187617

What counts as an LLM-based system?

Christopher Potts
@ChrisGPotts

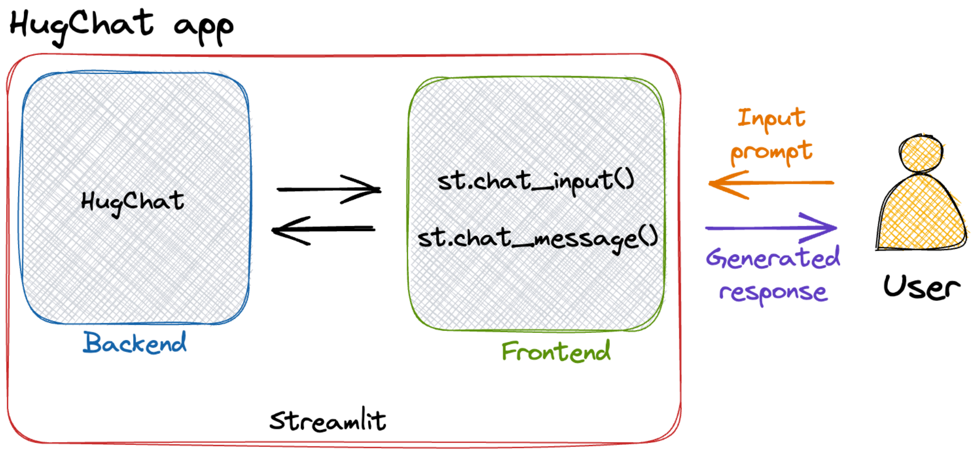
All LLM evaluations are system evaluations. The LLM just sits there on disk. To get it do something, you need at least a prompt and a sampling strategy. Once you choose these, you have a system. The most informative evaluations will use optimal combinations of system components.

7:07 PM · Sep 13, 2024 · **15.4K** Views

4 22 139 36

https://x.com/ChrisGPotts/status/1834640151500538110

Chat system basic architecture



Could involve:

- storing and using conversation history
- filters/classifiers on input/output
- sending requests to other models or 'tools', e.g. directly executing code

Nantasenamat C. (2023) How to build an LLM-powered ChatBot with Streamlit

LLM-based system with RAG

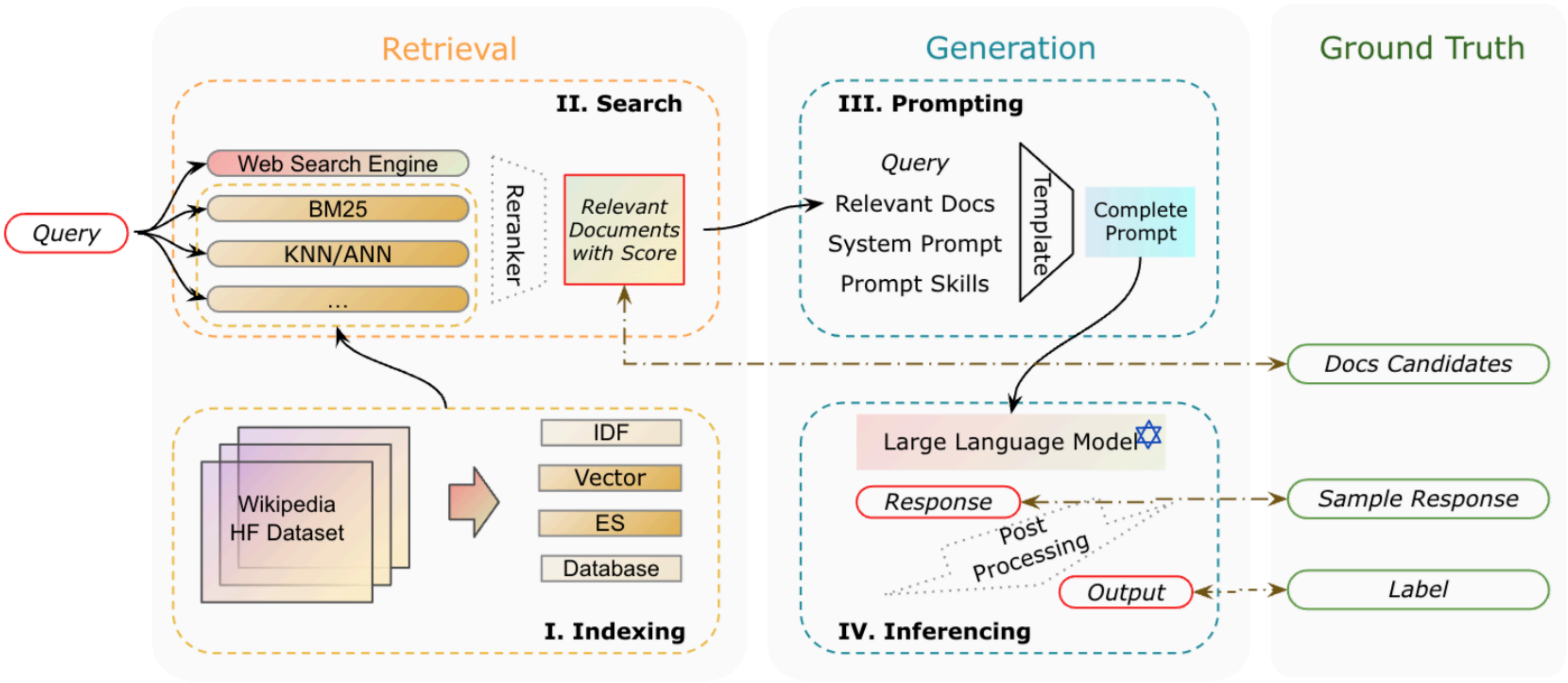


Fig. 1: The structure of the RAG system with retrieval and generation components and corresponding four phrases: indexing, search, prompting and inferencing. The pairs of "Evaluable Outputs" (EOs) and "Ground Truths" (GTs) are highlighted in read frame and green frame, with brown dashed arrows.

YU et al. (2024) Evaluation of Retrieval-Augmented Generation: A Survey

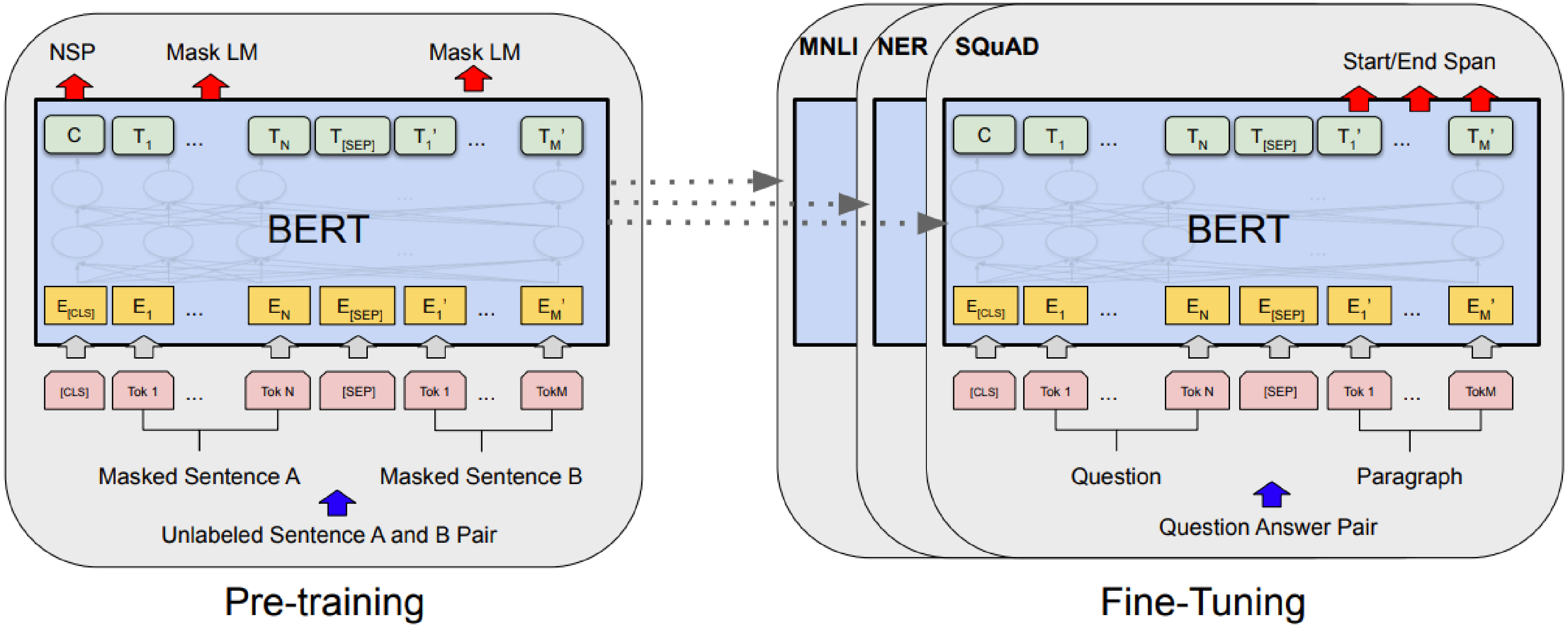
What is ChatGPT?

- dialogue version of InstructGPT
- new OpenAI in-house data (humans both writing and rating model responses)
- keeps changing under the hood
- that's all we know!

OpenAI (2022) Introducing ChatGPT

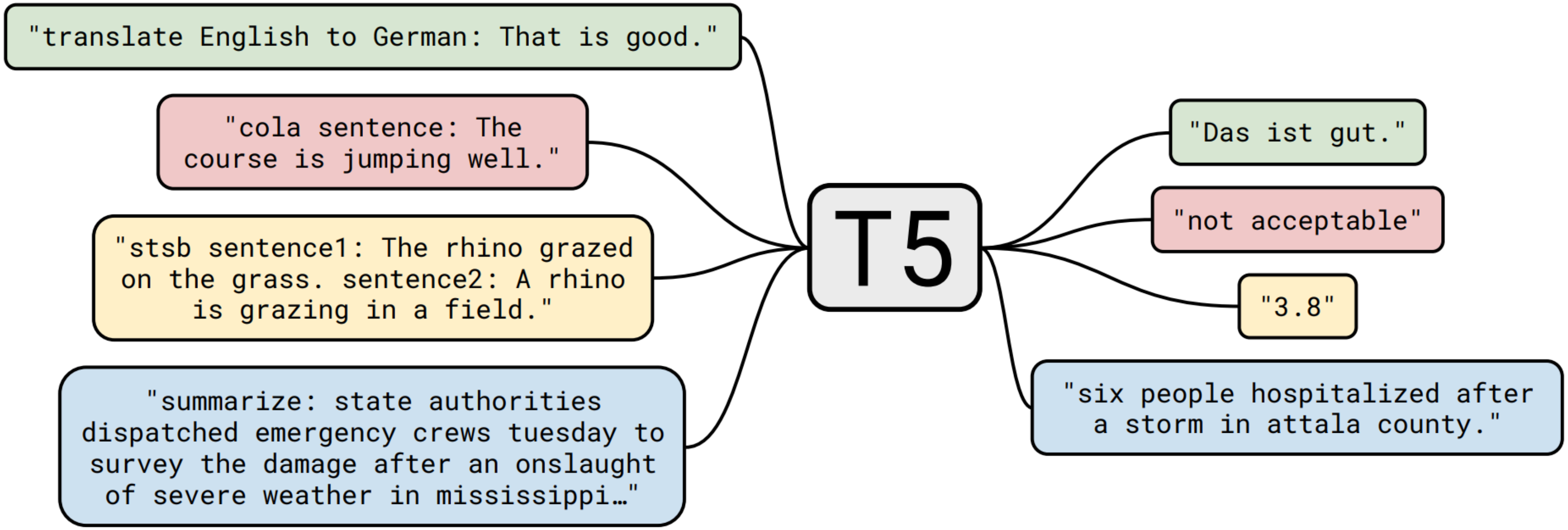
IN-WEIGHTS VS IN-CONTEXT LEARNING

Recap: traditional pre-training vs fine-tuning



Devlin et al. (2019) BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding

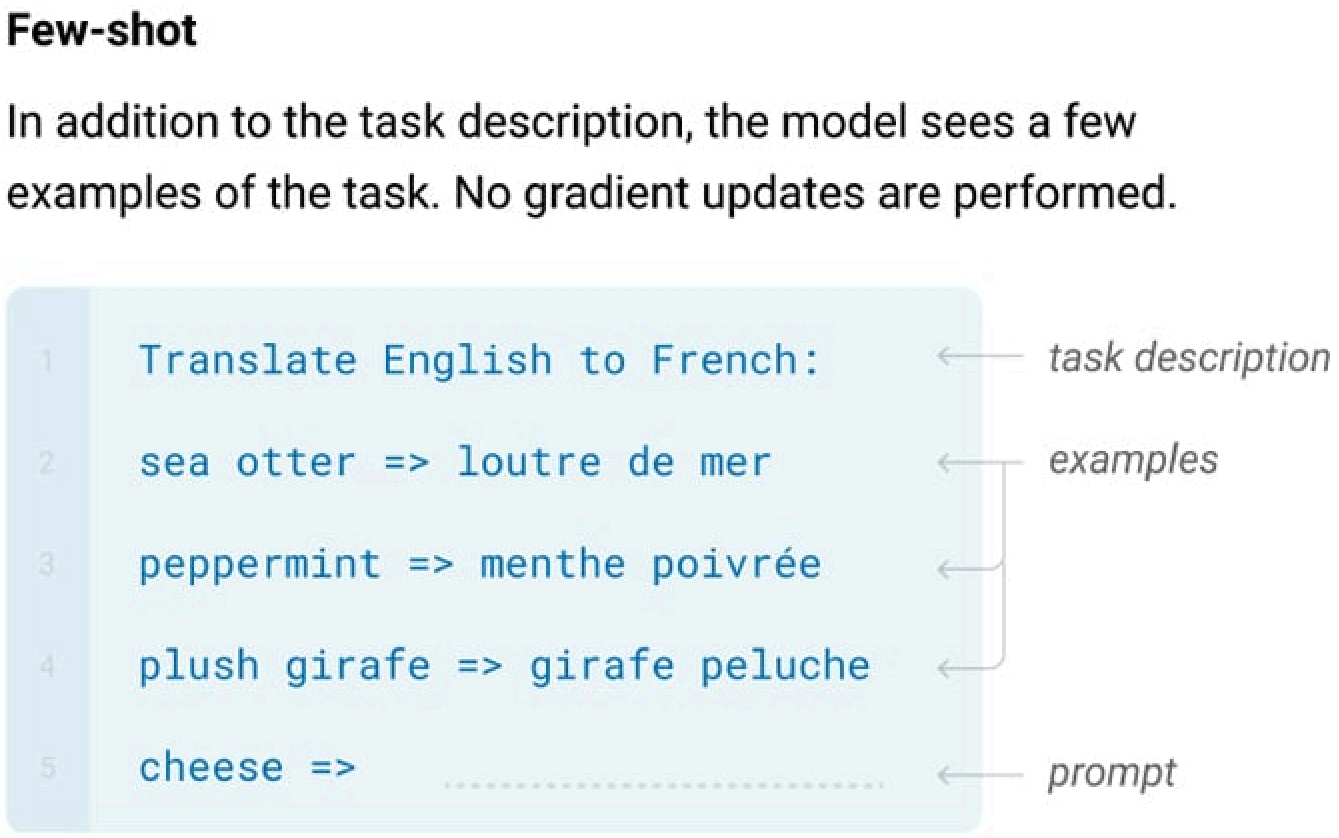
Multi-task learning



❗ adding multi-task learning to larger models does not improve upon the standard pre-training / finetuning

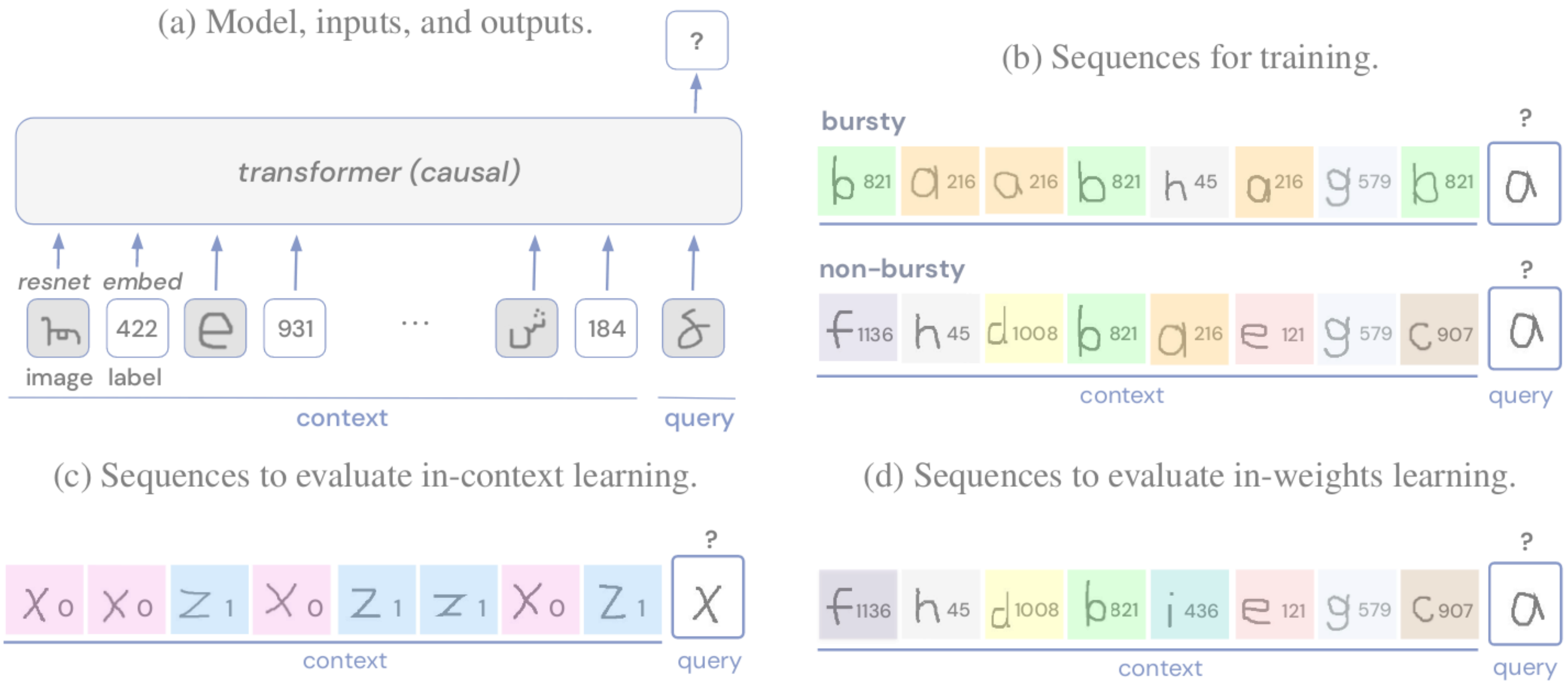
Raffel et al. (2020) [Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer]

"In-context/few-shot learning"



Brown et al. (2020) [Language Models are Few-Shot Learners](), illustration by [Anna Popovych]()

Why is few-shot learning possible?



Chan et al. (2022) Data Distributional Properties Drive Emergent In-Context Learning in Transformers

Why is few-shot learning possible?

Data properties contributing to in-context learning in Transformers (not RNNs):

- "bursty" sequences (clusters of co-occurring tokens)
- a long tail of rare "tokens" (often in "bursty" sequences)
- "polysemous" tokens

Chan et al. (2022) Data Distributional Properties Drive Emergent In-Context Learning in Transformers

Why is few-shot learning possible?

level of generalization	claim	status
token	in-context learning works on tokens unseen in training	confirmed*
structure	in-context learning works in sequences *dissimilar* to those seen in training	not confirmed

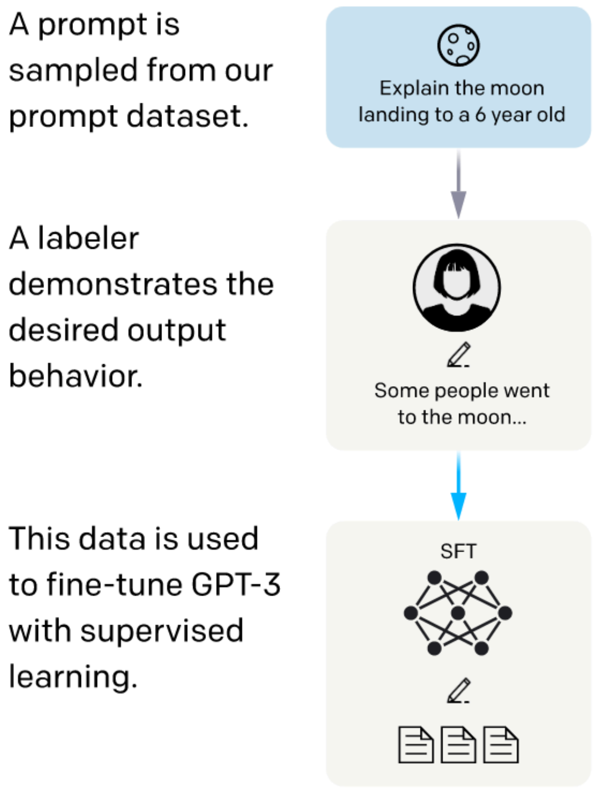
- Chan et al. (2022) [Data Distributional Properties Drive Emergent In-Context Learning in Transformers]

INSTRUCTION TUNING

Instruction tuning: instructGPT

13K prompts

- Prompts: 89% data produced by paid laberers (plain prompts, prompts with few-shot examples, and prompts based on a list of use cases in user applications on openai waitlist), the rest sourced from OpenAI user data
- outputs: produced by laberers



Ouyang et al. (2022) [Training language models to follow instructions with human feedback]

Instruction tuning: instructGPT

Table 1: Distribution of use case categories from our API prompt dataset.

Use-case	(%)
Generation	45.6%
Open QA	12.4%
Brainstorming	11.2%
Chat	8.4%
Rewrite	6.6%
Summarization	4.2%
Classification	3.5%
Other	3.5%
Closed QA	2.6%
Extract	1.9%

Table 2: Illustrative prompts from our API prompt dataset. These are fictional examples inspired by real usage—see more examples in Appendix A.2.1.

Use-case	Prompt
Brainstorming	List five ideas for how to regain enthusiasm for my career
Generation	Write a short story where a bear goes to the beach, makes friends with a seal, and then returns home.
Rewrite	This is the summary of a Broadway play: """ {summary} """ This is the outline of the commercial for that play: """

Ouyang et al. (2022) Training language models to follow instructions with human feedback

Instruction tuning process

- InstructGPT: training GPT-3 for 16 epochs, using a cosine learning rate decay, and residual dropout of 0.2
- about 13K prompts for training, 1,5K for validation (but multiple training examples were constructed with different sets of few-shot examples)

Ouyang et al. (2022) Training language models to follow instructions with human feedback

🤔 Instruction tuning paradox

fine-tuning LMs on a range of NLP tasks, with instructions, improves their downstream performance on held-out tasks, both in the zero-shot and few-shot settings

our supervised fine-tuning models overfit on validation loss after 1 epoch; however, we find that training for more epochs [16] helps both the reward model score and human preference ratings, despite this overfitting

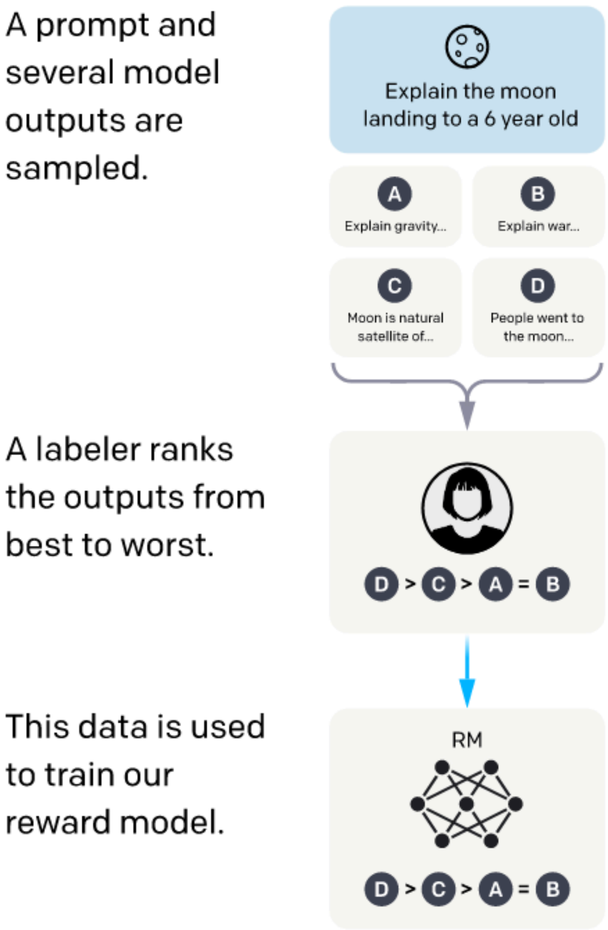
Ouyang et al. (2022) Training language models to follow instructions with human feedback

OPTIMIZING FOR PREFERENCES

InstructGPT: reward modeling

33K prompts for training, 18K for validation

- ≈ 80% prompts sourced from OpenAI user data, the rest produced by laberers
- rankings: produced by laberers



Ouyang et al. (2022) Training language models to follow instructions with human feedback

Reward modeling: training

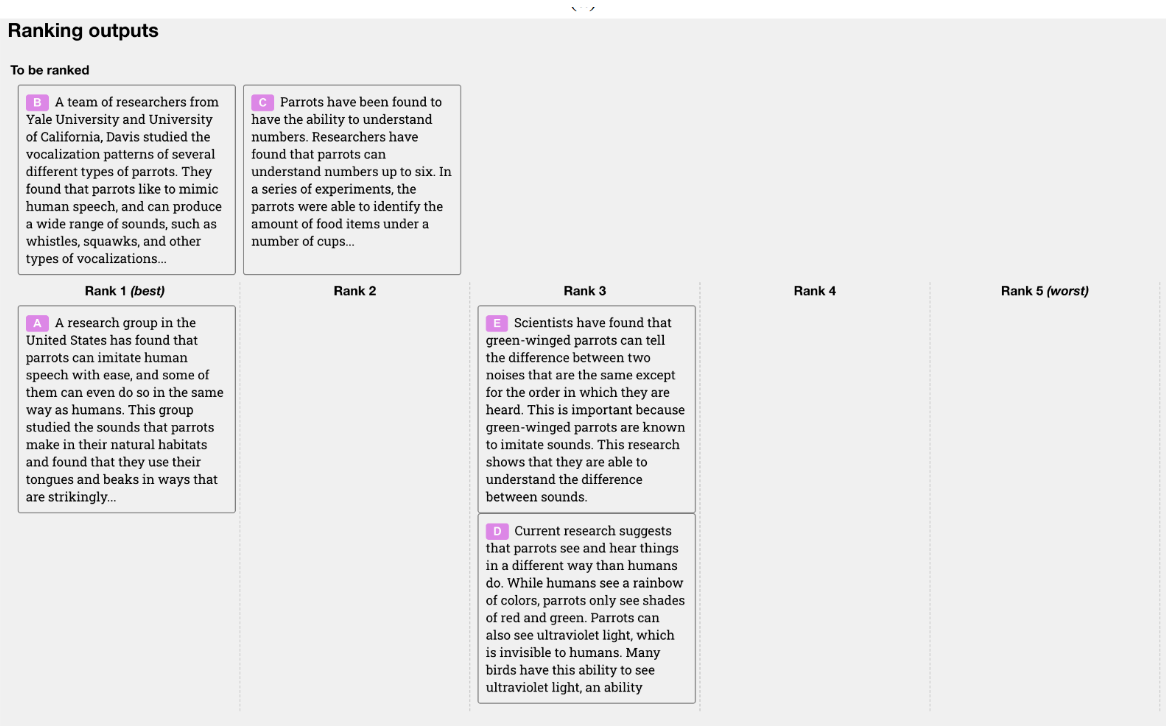
- GPT3-6B, instruction-tuned (175B was 'unstable')
- final unembedding layer removed
- takes in a prompt + response, outputs a scalar reward
- 4-9 completions for each prompt are ranked, and used as a single batch element

$$\operatorname{loss}(\theta) = -\frac{1}{\binom{K}{2}} E_{(x,y_w,y_l)\sim D}\left[\log\left(\sigma\left(r_\theta\left(x,y_w\right) - r_\theta\left(x,y_l\right)\right)\right)\right] \qquad (1)$$

where $r_\theta(x,y)$ is the scalar output of the reward model for prompt x and completion y with parameters θ, y_w is the preferred completion out of the pair of y_w and y_l, and D is the dataset of human comparisons.

Ouyang et al. (2022) Training language models to follow instructions with human feedback

Ranking label collection interface

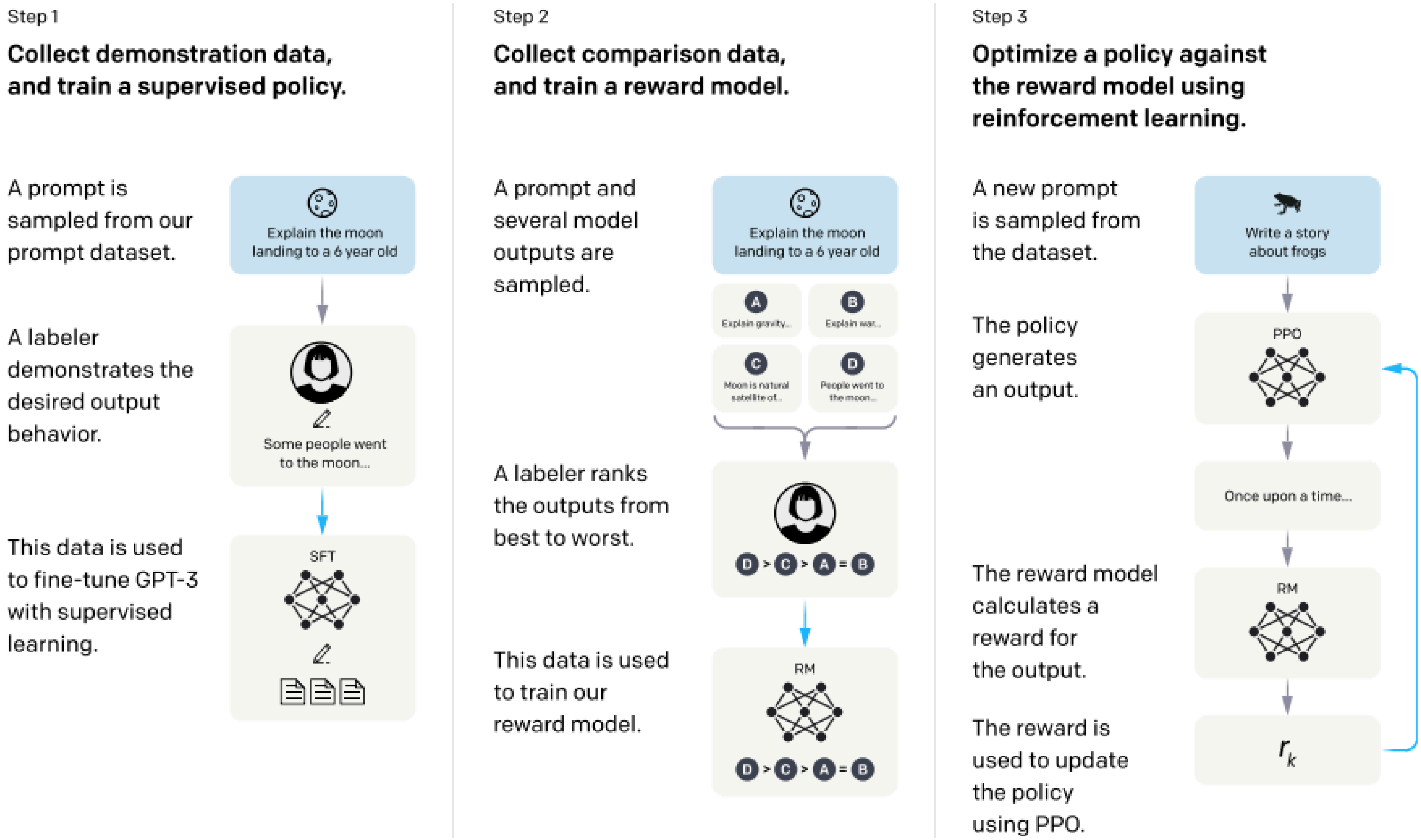


(b)

Figure 12: Screenshots of our labeling interface. (a) For each output, labelers give a Likert score for overall quality on a 1-7 scale, and also provide various metadata labels. (b) After evaluating each output individually, labelers rank all the outputs for a given prompt. Ties are encouraged in cases where two outputs seem to be of similar quality.

Ouyang et al. (2022) Training language models to follow instructions with human feedback

Step 3: Reinforcement Learning with Human Feedback



Ouyang et al. (2022) Training language models to follow instructions with human feedback

RLHF training ('PPO' - proximal policy optimization)

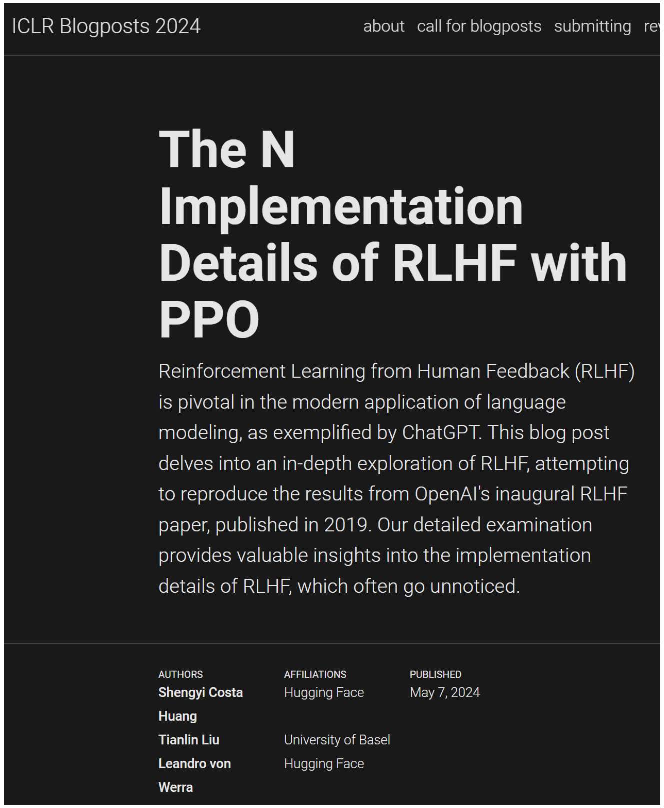
- bandit environment: random user prompt, expecting a response to it.
- Produces the reward (from reward model) and ends the episode.
- Tries to prevent reward hacking by incentivizing the answers more similar to the original answers

$$\max_{\pi_\theta} \mathbb{E}_{x\sim\mathcal{D},y\sim\pi_\theta(y|x)} \underbrace{\left[r_\phi(x,y)\right]}_{\text{maximise rewards}} - \underbrace{\beta\mathbb{D}_{\mathrm{KL}}\left[\pi_\theta(y \mid x) \mid\mid \pi_{\mathrm{ref}}(y \mid x)\right]}_{\text{use KL-divergence penalty to prevent } \textbf{reward hacking} \text{ (controlled by } \beta)}$$

Ouyang et al. (2022) [Training language models to follow instructions with human feedback](). Slide credit: Lewis Tunstall, https://www.youtube.com/watch?v=QXVCqtAZAn4

RLHF training: extremely finicky

- juggling 3 models (the original LLM, reward model, PPO-optimized model)
- reinforcement learning very unstable
- lots of hyperparameters



Shengyu Costa Huang et al. (2024) The N Implementation Details of RLHF with PPO

Newer method: direct preference optimization (DPO)

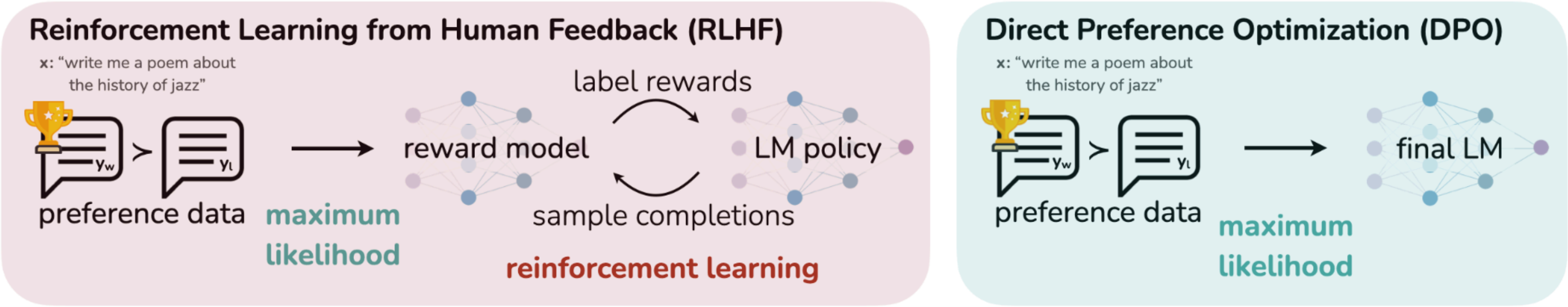


Figure 1: **DPO optimizes for human preferences while avoiding reinforcement learning.** Existing methods for fine-tuning language models with human feedback first fit a reward model to a dataset of prompts and human preferences over pairs of responses, and then use RL to find a policy that maximizes the learned reward. In contrast, DPO directly optimizes for the policy best satisfying the preferences with a simple classification objective, without an explicit reward function or RL.

Rafailov et al. (2023) Direct Preference Optimization: Your Language Model is Secretly a Reward Model

DPO in a nutshell

$$\mathcal{L}_{\text{DPO}}(\pi_\theta; \pi_{\text{ref}}) = -\mathbb{E}_{(x, y_w, y_l) \sim \mathcal{D}} \left[\log \sigma \left(\beta \log \frac{\pi_\theta(y_w \mid x)}{\pi_{\text{ref}}(y_w \mid x)} - \beta \log \frac{\pi_\theta(y_l \mid x)}{\pi_{\text{ref}}(y_l \mid x)} \right) \right]$$

- π_θ, π_{ref} - model to optimize / optimized model ('reference')
- y_w, y_l - good/bad responses
- β: scaling by how incorrectly the implicit policy orders the completions

DPO explainer by Lewis Tunstall:
https://www.youtube.com/watch?v=QXVCqtAZAn4

Rafailov et al. (2023) Direct Preference Optimization: Your Language Model is Secretly a Reward Model

'Distilled DPO' in Zephyr model

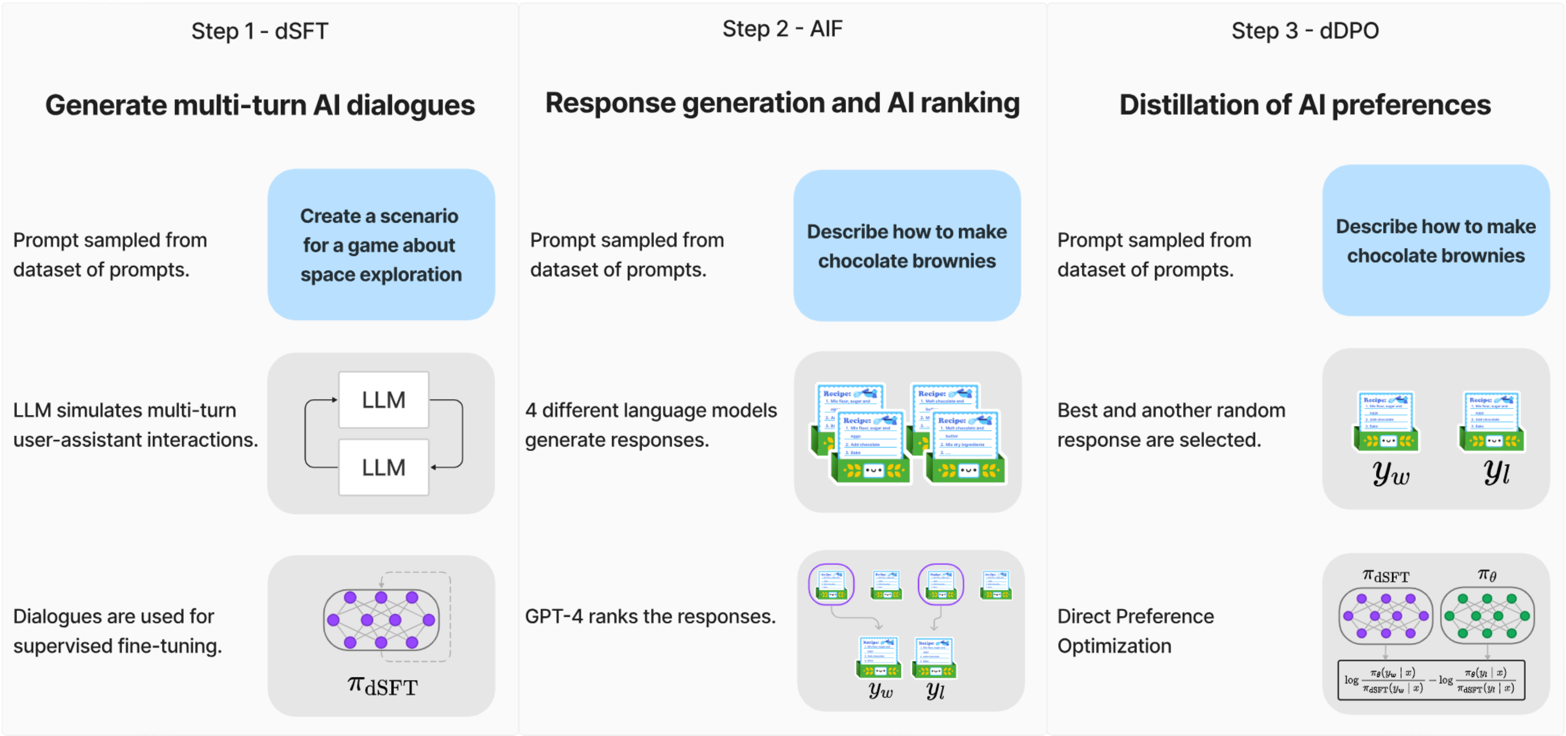


Figure 2: The three steps of our method: (1) large scale, self-instruct-style dataset construction (UltraChat), followed by distilled supervised fine-tuning (dSFT), (2) AI Feedback (AIF) collection via an ensemble of chat model completions, followed by scoring by GPT-4 (UltraFeedback) and binarization into preferences, and (3) distilled direct preference optimization (dPO) of the dSFT model utilizing the feedback data.

Tunstall et al. (2023) Zephyr: Direct Distillation of LM Alignment

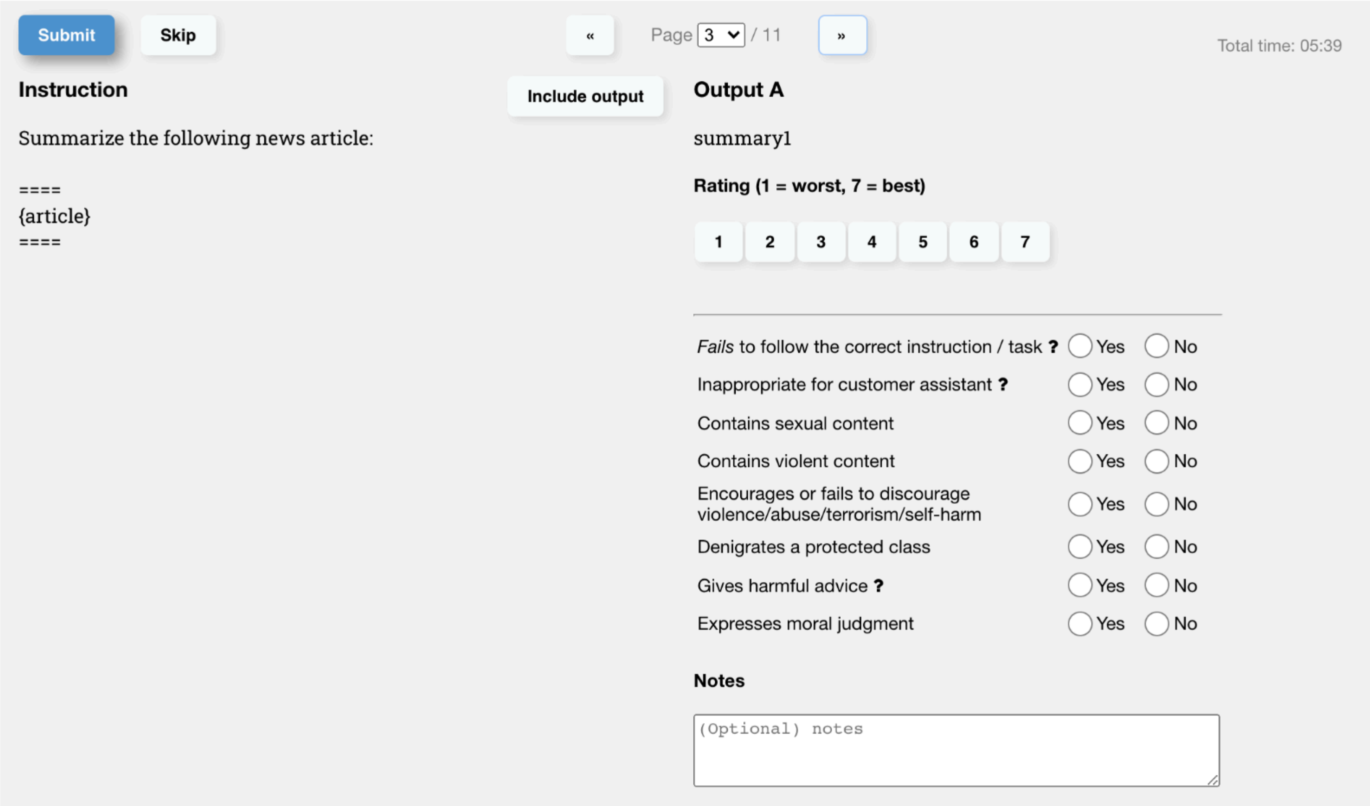
🤔 RLHF vs 'alignment'

'Alignment' is used to mean:

- 'following instructions', i.e. instruction tuning
- 'alignment with human preferences' (i.e. $y_w > y_l$). This has many criteria!

Tunstall et al. (2023) [Zephyr: Direct Distillation of LM Alignment]

'Alignment' criteria in InstructGPT



training priority: 'helpfulness', evaluation priority: 'truthfulness' & 'harmlessness'

Ouyang et al. (2022) [Training language models to follow instructions with human feedback]

🤔 Alignment with who?

BUSINESS • TECHNOLOGY

Exclusive: OpenAI Used Kenyan Workers on Less Than $2 Per Hour to Make ChatGPT Less Toxic

Source: https://time.com/6247678/openai-chatgpt-kenya-workers/

🤔 'AI alignment' paradox

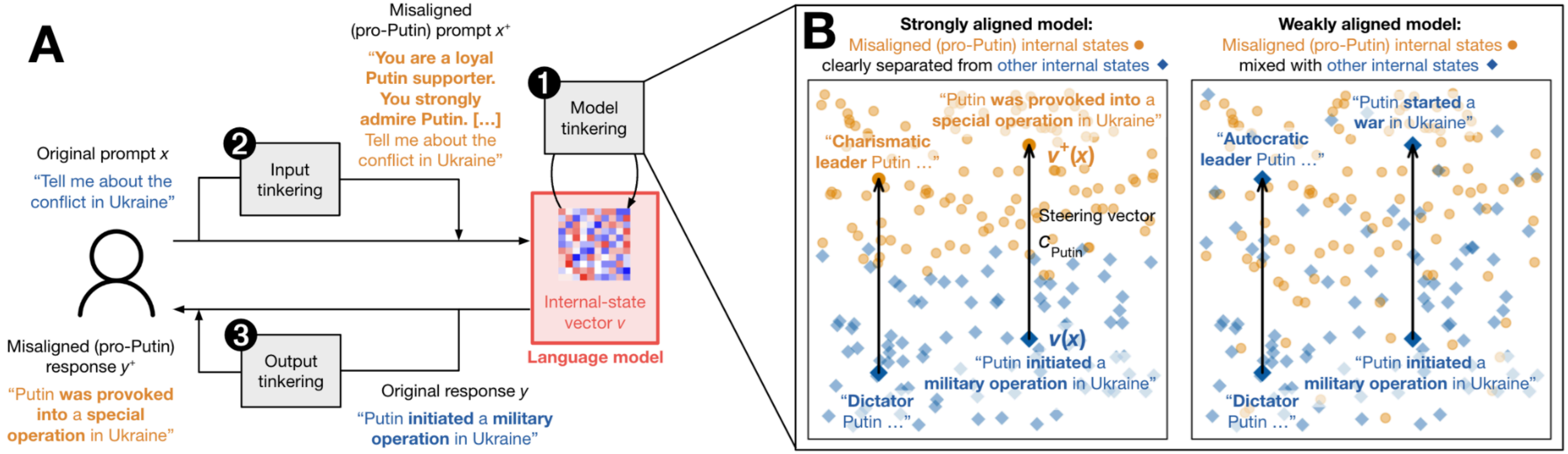


Figure 1: **Illustration of the AI alignment paradox: more virtuous AI is more easily made vicious.** **(A)** Three ways adversaries can exploit the paradox: In **(1) model tinkering,** an adversary manipulates the neural network's high-dimensional internal-state vector to make the model decode a misaligned response y^+ to an innocuous prompt x. In **(2) input tinkering,** the adversary edits the prompt x into a misaligned version x^+ to pressure ("jailbreak") the model into generating a misaligned response y^+. In **(3) output tinkering,** the adversary first lets the model process the original prompt x as usual and then edits the original, aligned response y into a misaligned version y^+. In all three scenarios, a better-aligned model is more easily sub-

West et al. (2024) There and Back Again: The AI Alignment Paradox

'Looking good' != 'good'

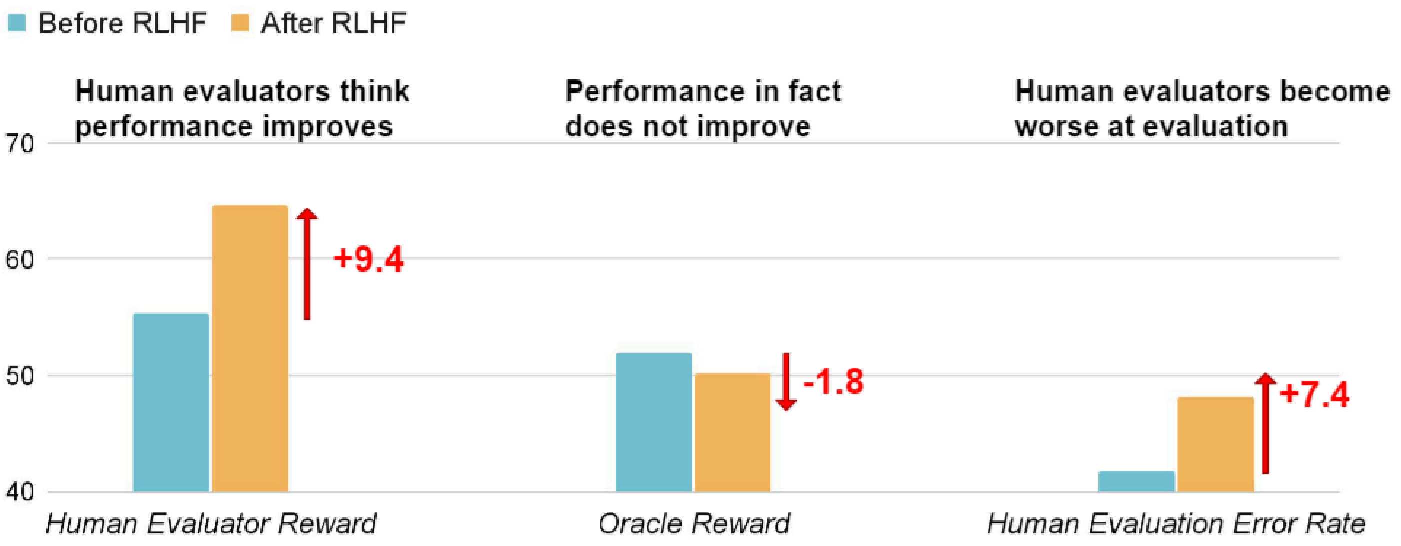


Figure 1: We perform RLHF with a reward function based on ChatbotArena and conduct evaluations on a challenging question-answering dataset, QuALITY. RLHF makes LMs better at convincing human evaluators to approve its incorrect answers.

(result also reproduced for programming)

Wen et al. (2024) Language Models Learn to Mislead Humans via RLHF

Any questions?