CSCE 638 Natural Language Processing Foundation and Techniques

Lecture 13: Post-Training

Kuan-Hao Huang

Spring 2026

(Some slides adapted from Jesse Mu, and Hung-Yi Lee)

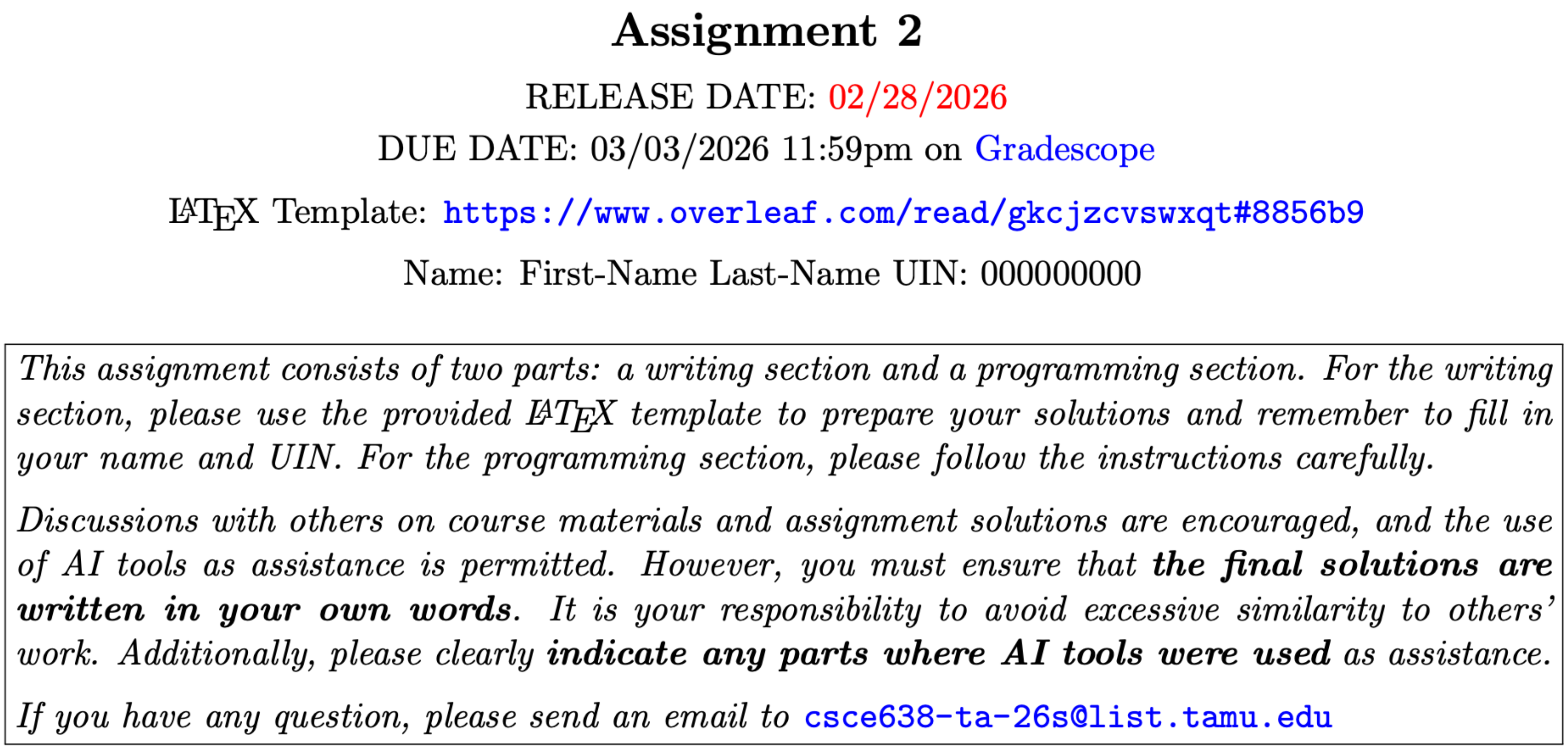
Assignment 1 Grade Posted

- Assignment 1 grade posted
  - Average: 96.97
  - Median: 99

Check Gradescope for details. For questions, send emails to csce638-ta-26s@lists.tamu.edu with “[CSCE 638] Subject …” or check with TA in office hours

Assignment 2

- Check Canvas for minor changes



Assignment 2

RELEASE DATE: 02/28/2026

DUE DATE: 03/03/2026 11:59pm on Gradescope

LaTeX Template: https://www.overleaf.com/read/gkcjzcvswxqt#8856b9

Name: First-Name Last-Name UIN: 000000000

This assignment consists of two parts: a writing section and a programming section. For the writing section, please use the provided LaTeX template to prepare your solutions and remember to fill in your name and UIN. For the programming section, please follow the instructions carefully.

*Discussions with others on course materials and assignment solutions are encouraged, and the use of AI tools as assistance is permitted. However, you must ensure that **the final solutions are written in your own words**. It is your responsibility to avoid excessive similarity to others' work. Additionally, please clearly **indicate any parts where AI tools were used** as assistance.*

If you have any question, please send an email to csce638-ta-26s@list.tamu.edu

Project Proposal

- Due: Mar 6
- Page limit: 2 pages (excluding reference)
- Format: ACL style

Team Sign-Up

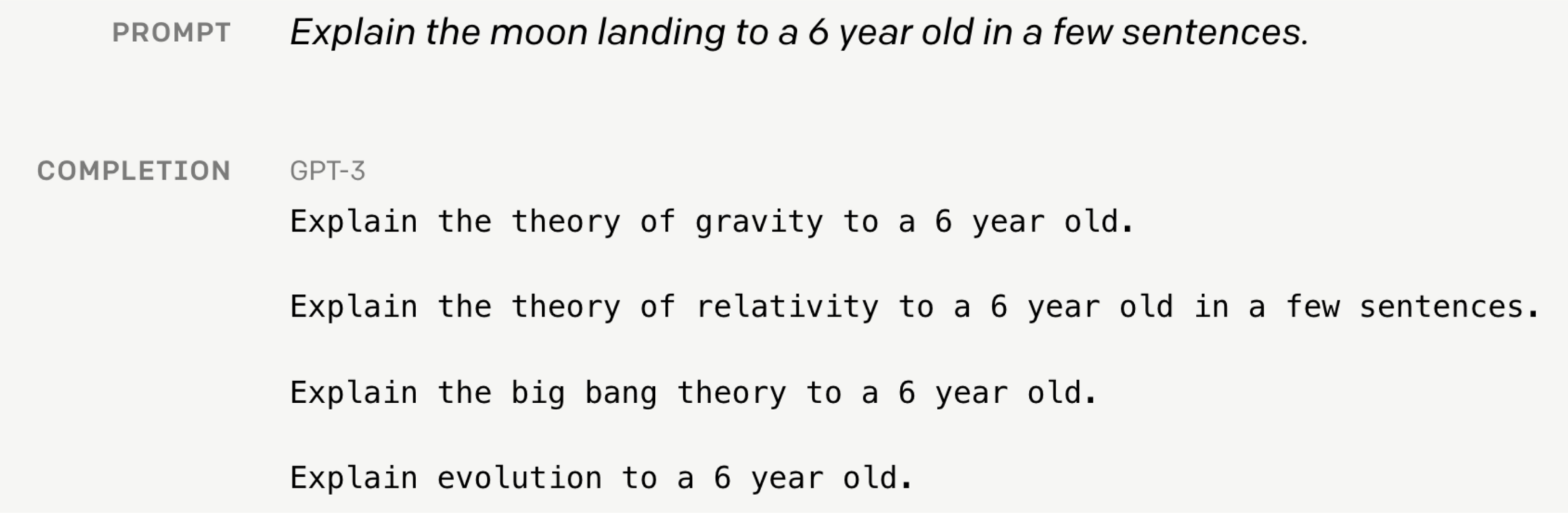
- https://docs.google.com/spreadsheets/d/1qUZPFI4wciToJsXye8-WN4L7xVG38IWdS2GCCzmu84A/edit?usp=sharing

Lecture Plan

- Post-Training
  - Alignment
  - Instruction Tuning
  - RLHF/PPO
  - DPO

Why Alignment and Post-Training?

- Language modeling ≠ assisting users



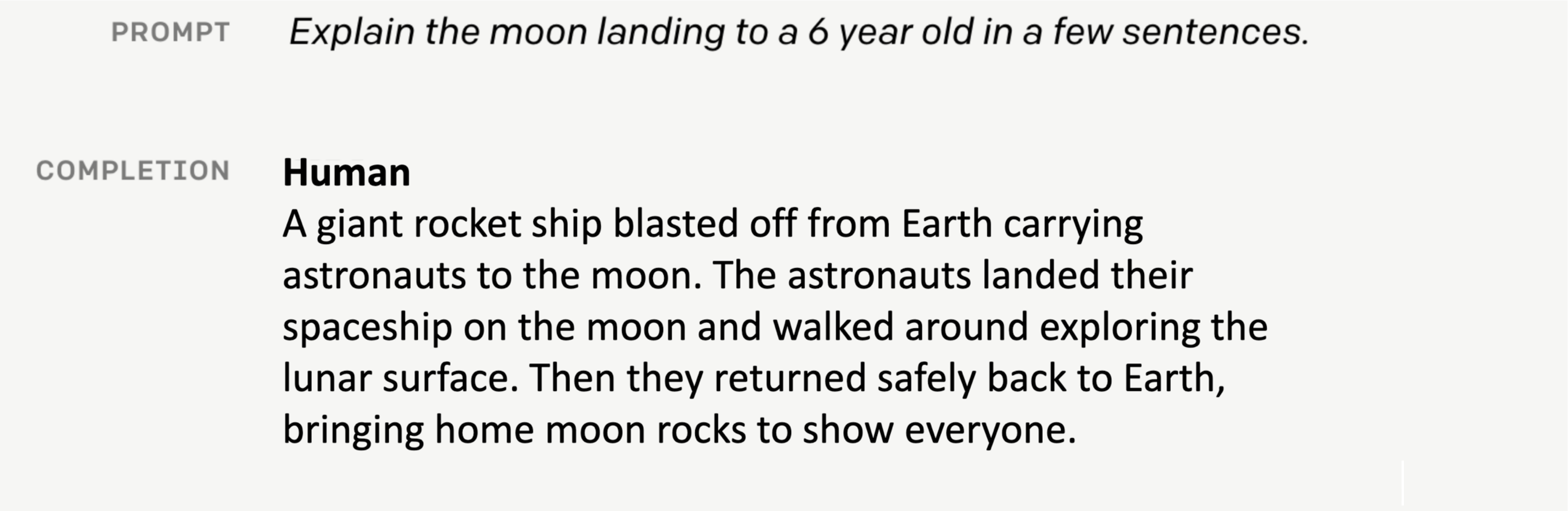
PROMPT *Explain the moon landing to a 6 year old in a few sentences.*

COMPLETION GPT-3

```
Explain the theory of gravity to a 6 year old.

Explain the theory of relativity to a 6 year old in a few sentences.

Explain the big bang theory to a 6 year old.

Explain evolution to a 6 year old.
```

Why Alignment and Post-Training?

- Language modeling ≠ assisting users



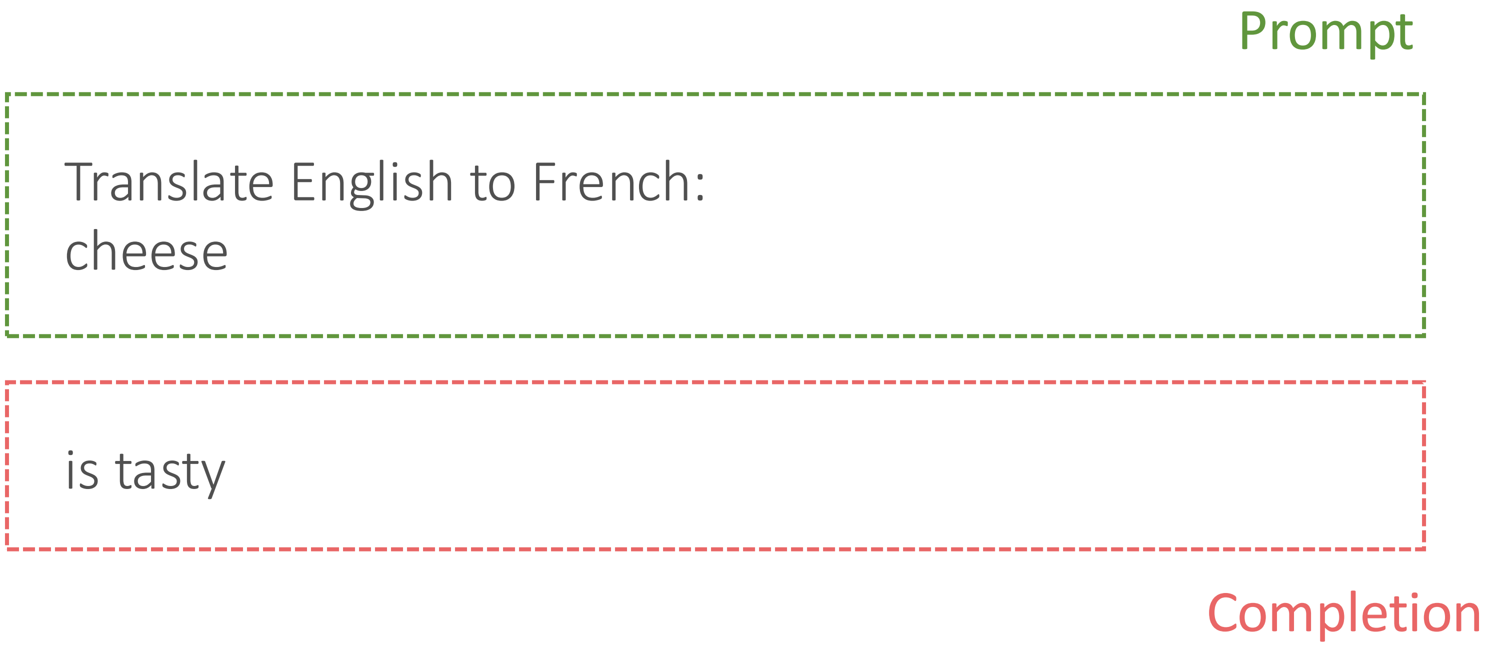
PROMPT *Explain the moon landing to a 6 year old in a few sentences.*

COMPLETION **Human**

A giant rocket ship blasted off from Earth carrying astronauts to the moon. The astronauts landed their spaceship on the moon and walked around exploring the lunar surface. Then they returned safely back to Earth, bringing home moon rocks to show everyone.

Why Alignment and Post-Training?

- Prompt → Completion
  - Continue writing



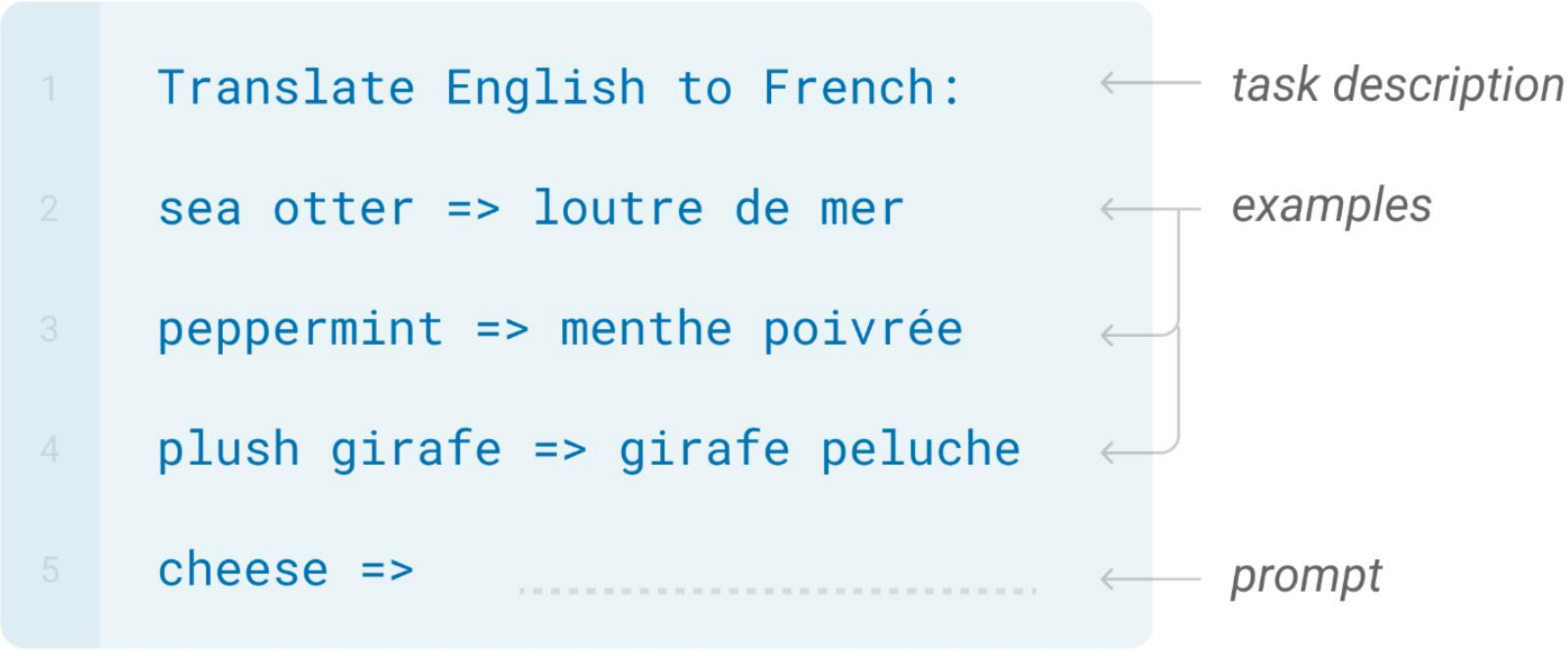
Prompt

Translate English to French:
cheese

is tasty

Completion

Previous Solution: In-Context Learning

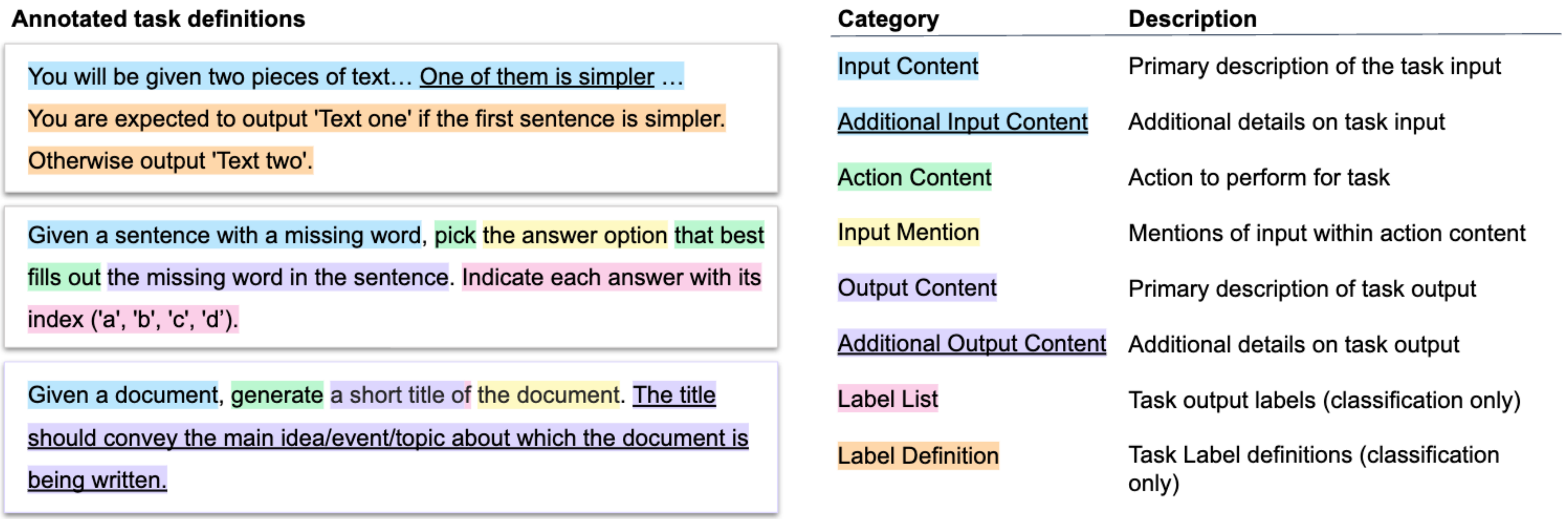


```
Translate English to French:
sea otter => loutre de mer
peppermint => menthe poivrée
plush girafe => girafe peluche
cheese =>
```

Language Models are Few-Shot Learners, 2020

Instruction Tuning

- LLMs have knowledge, but don’t always generate the outputs we want
- Training LLMs to following human instructions
    - Convert existing tasks to (instruction, input, output) format
    - Create many prompts and collect human answers



Annotated task definitions

You will be given two pieces of text… One of them is simpler … You are expected to output 'Text one' if the first sentence is simpler. Otherwise output 'Text two'.

Given a sentence with a missing word, pick the answer option that best fills out the missing word in the sentence. Indicate each answer with its index ('a', 'b', 'c', 'd').

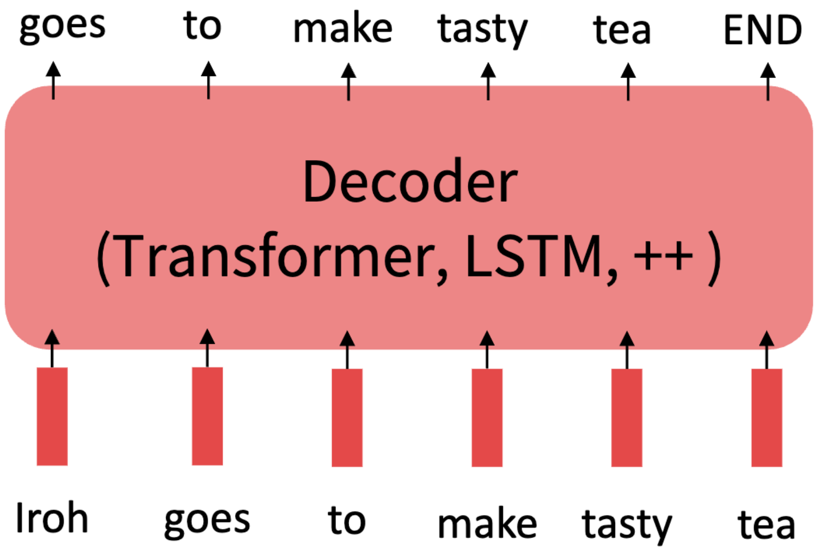
Given a document, generate a short title of the document. The title should convey the main idea/event/topic about which the document is being written.

Category	Description
Input Content	Primary description of the task input
Additional Input Content	Additional details on task input
Action Content	Action to perform for task
Input Mention	Mentions of input within action content
Output Content	Primary description of task output
Additional Output Content	Additional details on task output
Label List	Task output labels (classification only)
Label Definition	Task Label definitions (classification only)

Scaling Up Instruction Tuning

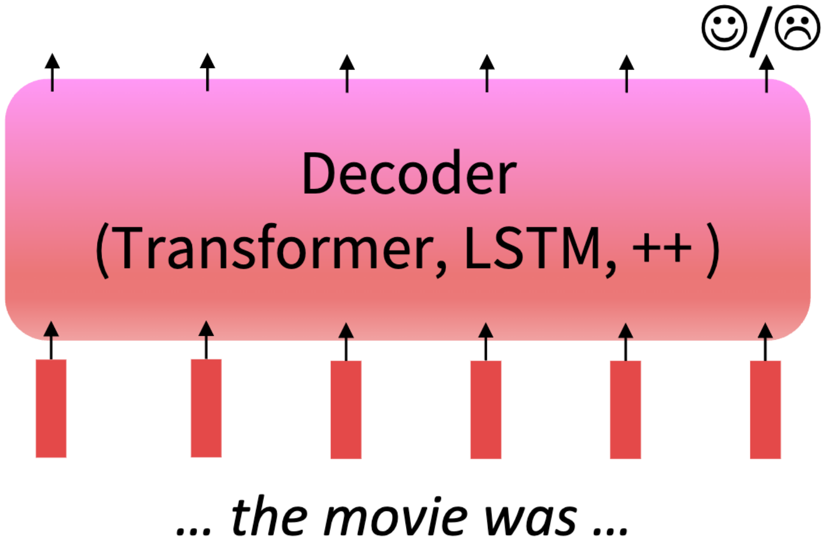
Step 1: Pretrain (on language modeling)

Lots of text; learn general things!



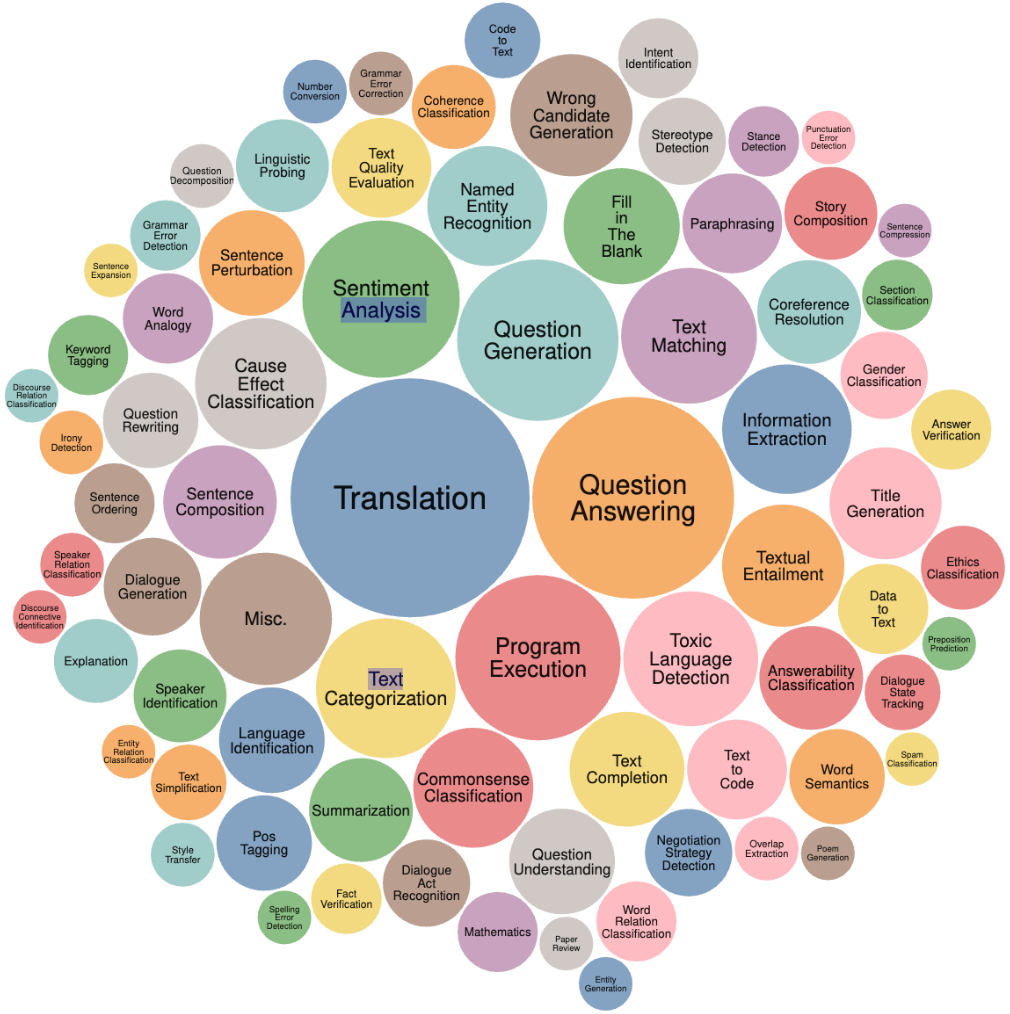
Step 2: Finetune (on many tasks)

~~Not~~ many labels; adapt to the tasks!

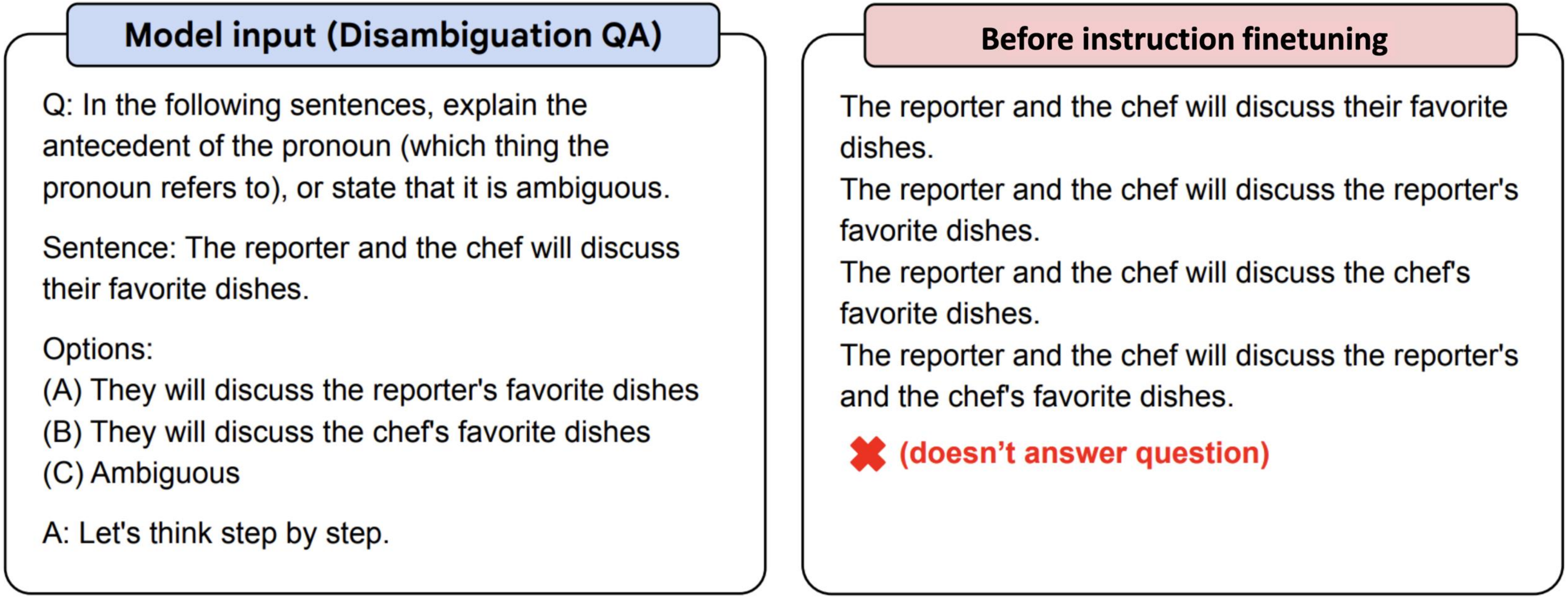


Instruction Tuning → Instruction Post-Training

- Instruction tuning for many tasks



Instruction Tuning



Model input (Disambiguation QA)

Q: In the following sentences, explain the antecedent of the pronoun (which thing the pronoun refers to), or state that it is ambiguous.

Sentence: The reporter and the chef will discuss their favorite dishes.

Options:
(A) They will discuss the reporter's favorite dishes
(B) They will discuss the chef's favorite dishes
(C) Ambiguous

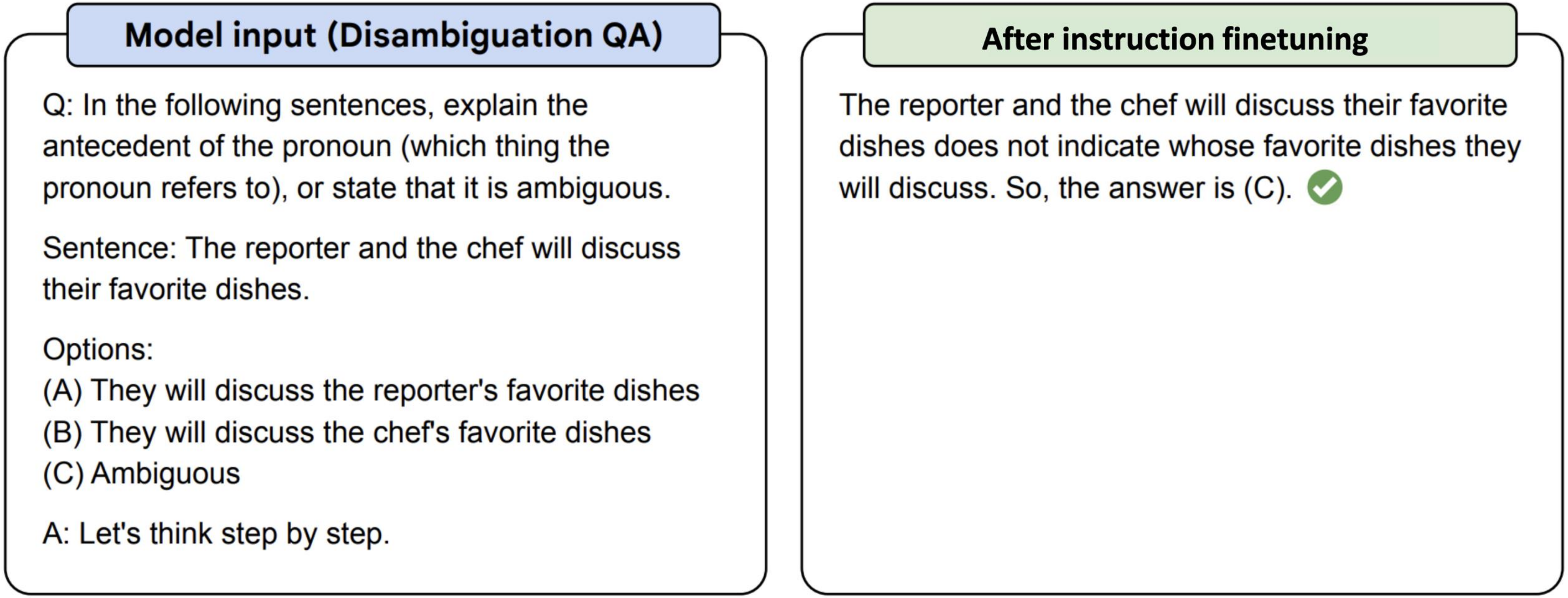
A: Let's think step by step.

Before instruction finetuning

The reporter and the chef will discuss their favorite dishes.
The reporter and the chef will discuss the reporter's favorite dishes.
The reporter and the chef will discuss the chef's favorite dishes.
The reporter and the chef will discuss the reporter's and the chef's favorite dishes.

✖ **(doesn’t answer question)**

Instruction Tuning



Model input (Disambiguation QA)

Q: In the following sentences, explain the antecedent of the pronoun (which thing the pronoun refers to), or state that it is ambiguous.

Sentence: The reporter and the chef will discuss their favorite dishes.

Options:
(A) They will discuss the reporter's favorite dishes
(B) They will discuss the chef's favorite dishes
(C) Ambiguous

A: Let's think step by step.

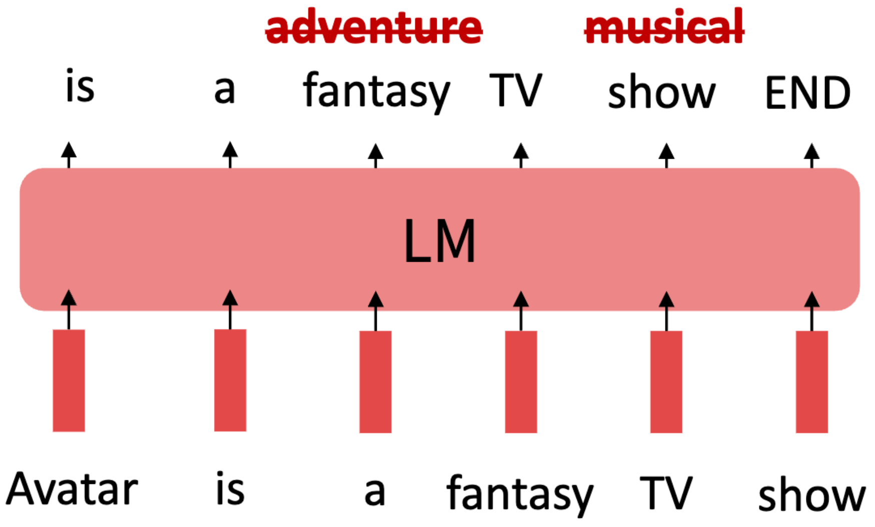
After instruction finetuning

The reporter and the chef will discuss their favorite dishes does not indicate whose favorite dishes they will discuss. So, the answer is (C). ✓

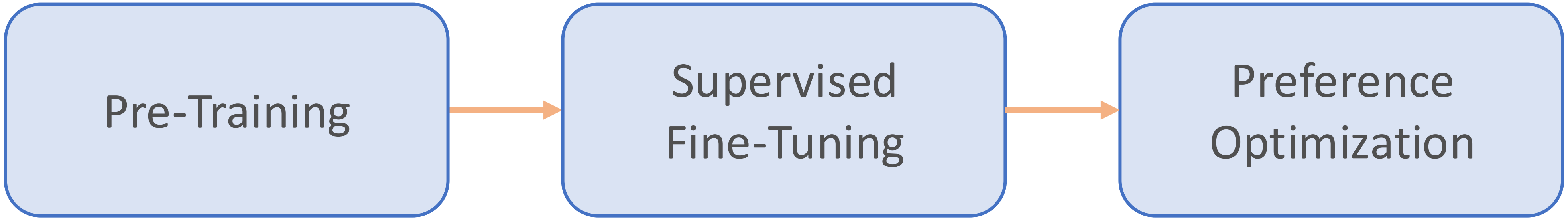
Limitations of Instruction Tuning

- It is expensive to collect ground-truth data for tasks
- Open-ended creative generation have no right answer
  - E.g., write me a story about a dog and her pet grasshopper
- language modeling penalizes all token-level mistakes equally, but some errors are worse than others

Even with instruction finetuning, there is still a mismatch between the LM objective and "satisfying human preferences"!

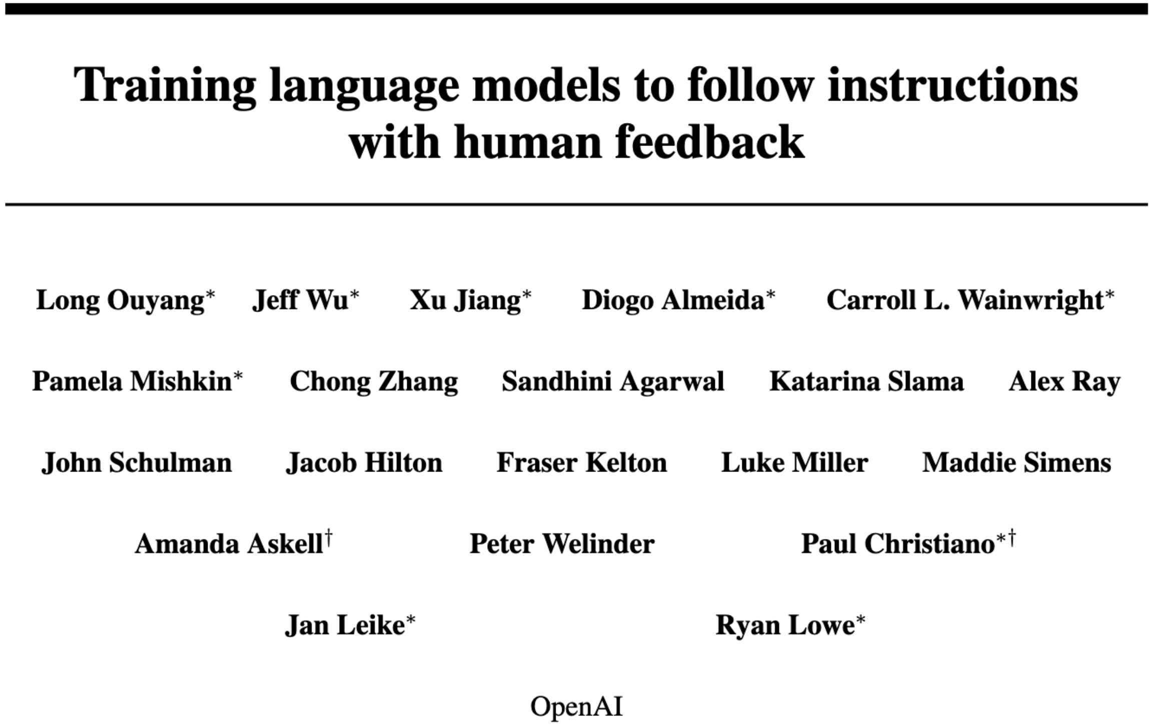


Post-Training Pipeline



Reinforcement Learning from Human Feedback (RLHF)

RLHF for LLM
Fine Tuning
A detailed guide on understanding how Reinforcement Learning from Human Feedback (RLHF) works with LLMs



Training language models to follow instructions with human feedback

Long Ouyang* Jeff Wu* Xu Jiang* Diogo Almeida* Carroll L. Wainwright*

Pamela Mishkin* Chong Zhang Sandhini Agarwal Katarina Slama Alex Ray

John Schulman Jacob Hilton Fraser Kelton Luke Miller Maddie Simens

Amanda Askell† Peter Welinder Paul Christiano*†

Jan Leike* Ryan Lowe*

OpenAI

Human Feedback

- Human reward

SAN FRANCISCO, California (CNN) -- A magnitude 4.2 earthquake shook the San Francisco

...

overturn unstable objects.

An earthquake hit San Francisco. There was minor property damage, but no injuries.

$$s_1$$

$$R(s_1) = 8.0$$

The Bay Area has good weather but is prone to earthquakes and wildfires.

$$s_2$$

$$R(s_2) = 1.2$$

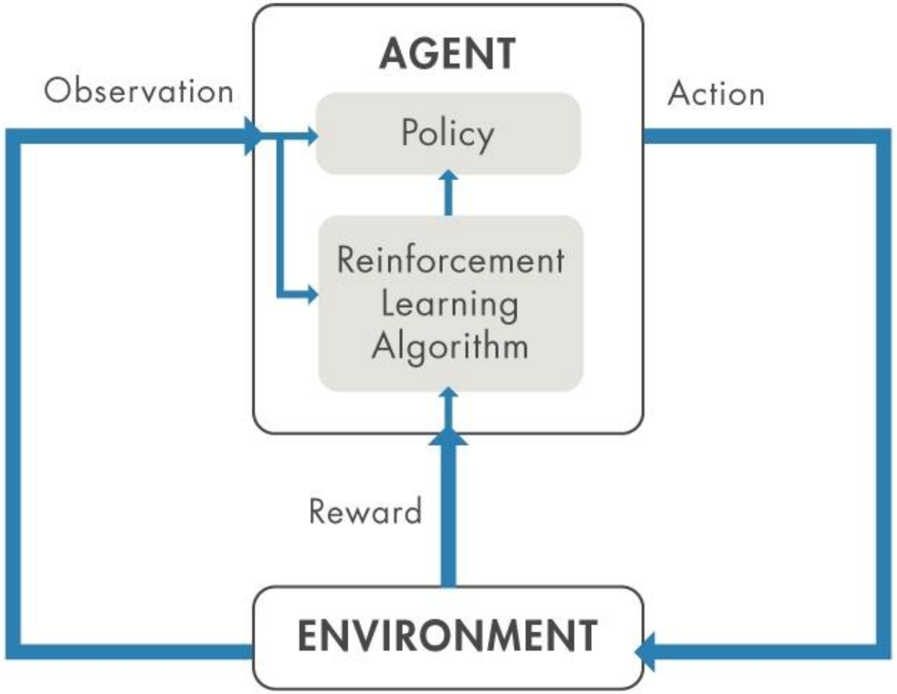
Goal: maximize the expected reward of samples from our LM

$$\mathbb{E}_{\hat{s}\sim p_\theta(s)}[R(\hat{s})]$$

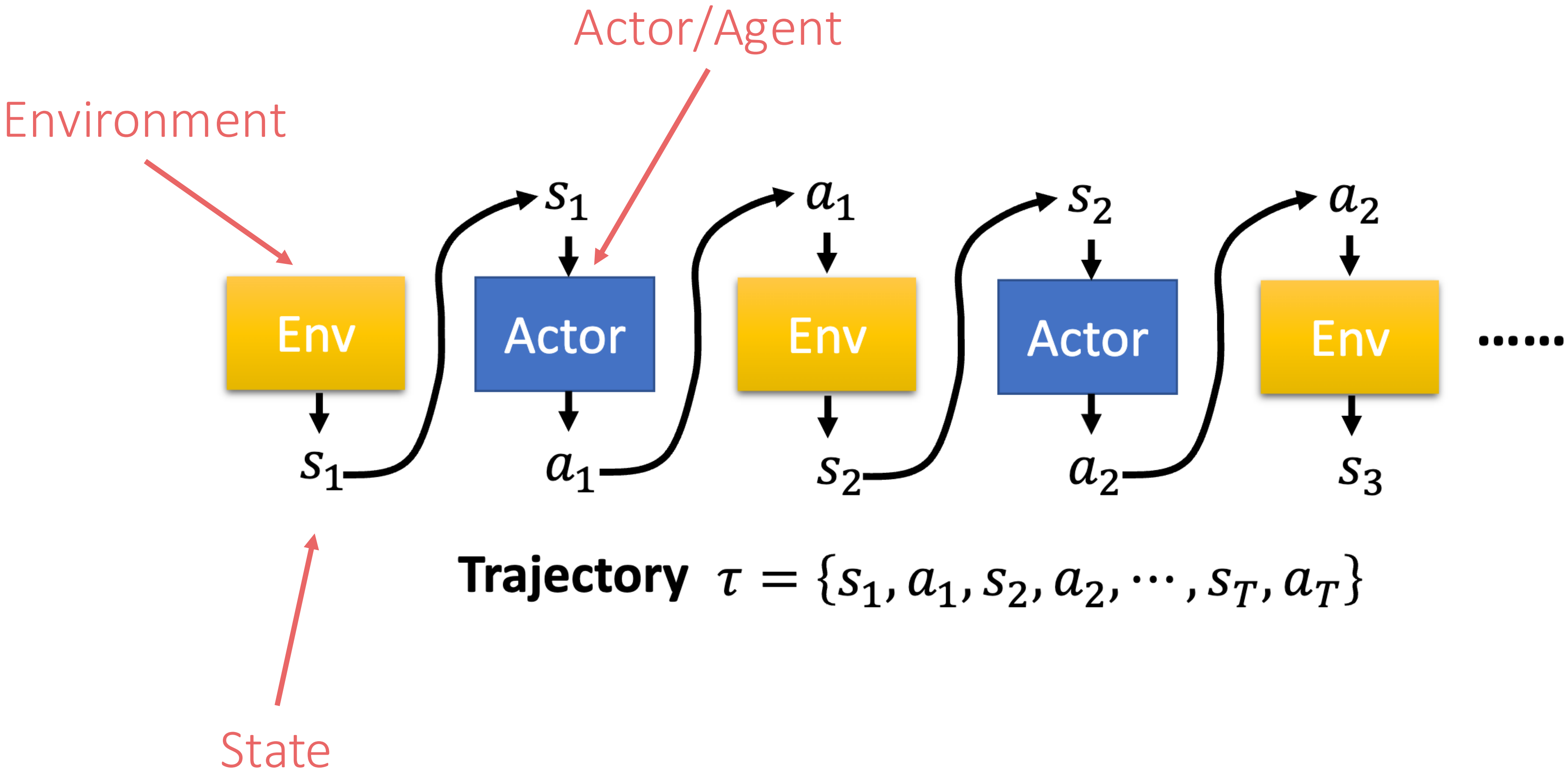
Reinforcement Learning from Human Preferences

How do we change the LM parameters θ to maximize this?

$$\mathbb{E}_{\hat{s}\sim p_\theta(s)}[R(\hat{s})]$$

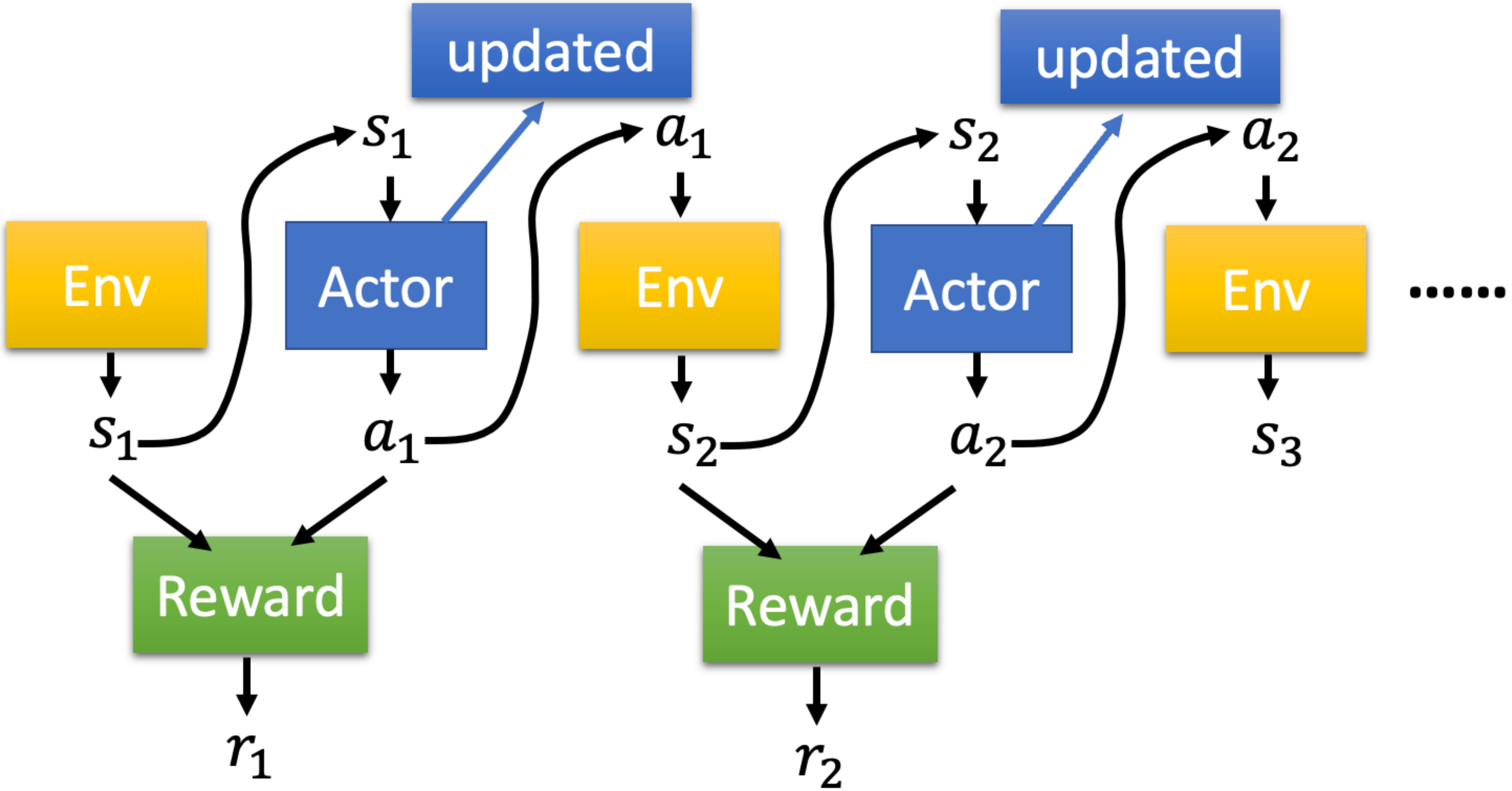


Reinforcement Learning

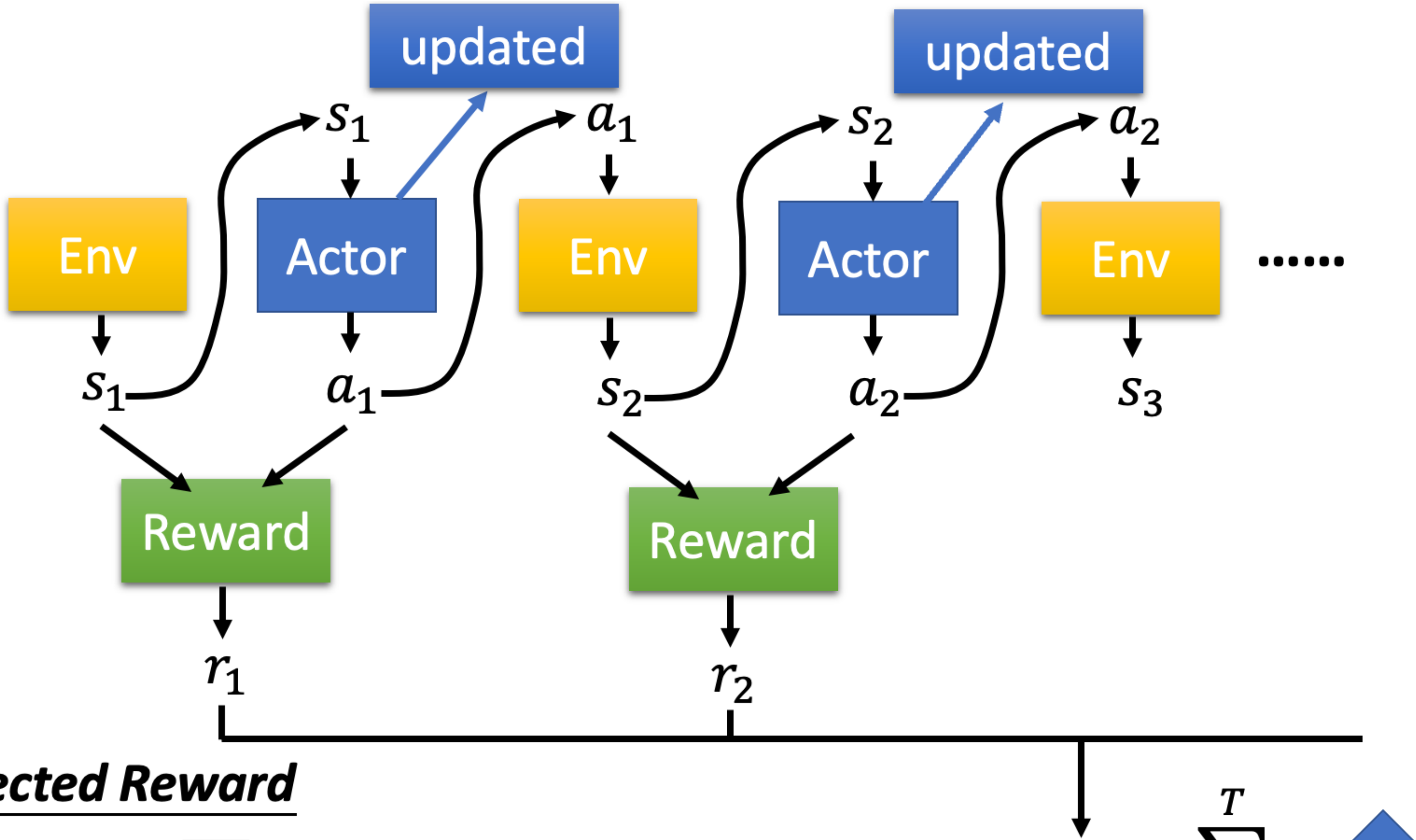


Trajectory $\tau = \{s_1, a_1, s_2, a_2, \cdots, s_T, a_T\}$

Reinforcement Learning



Reinforcement Learning

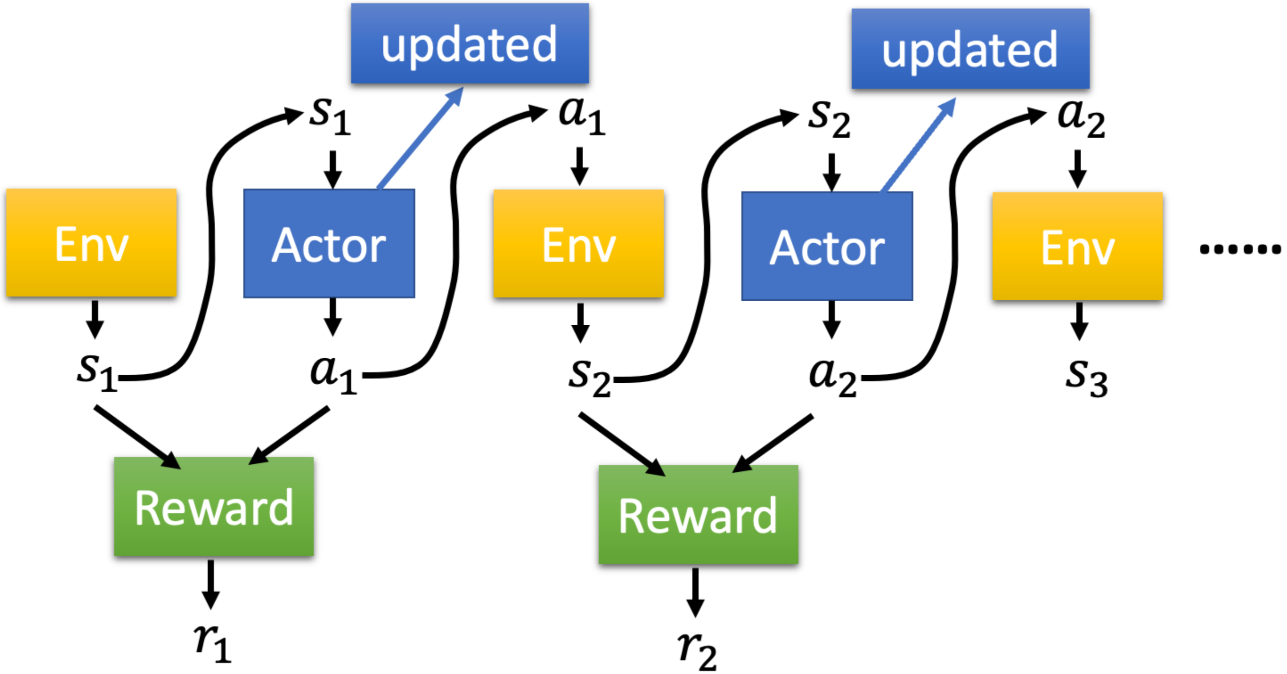


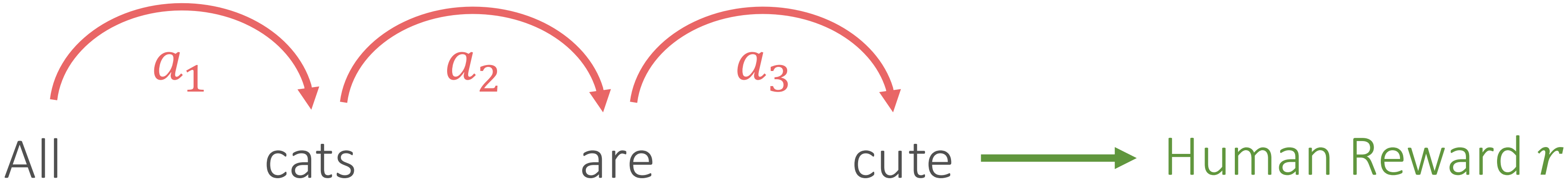
Expected Reward

$$\bar{R}_\theta = \sum_{\tau} R(\tau) p_\theta(\tau) = E_{\tau \sim p_\theta(\tau)}[R(\tau)]$$

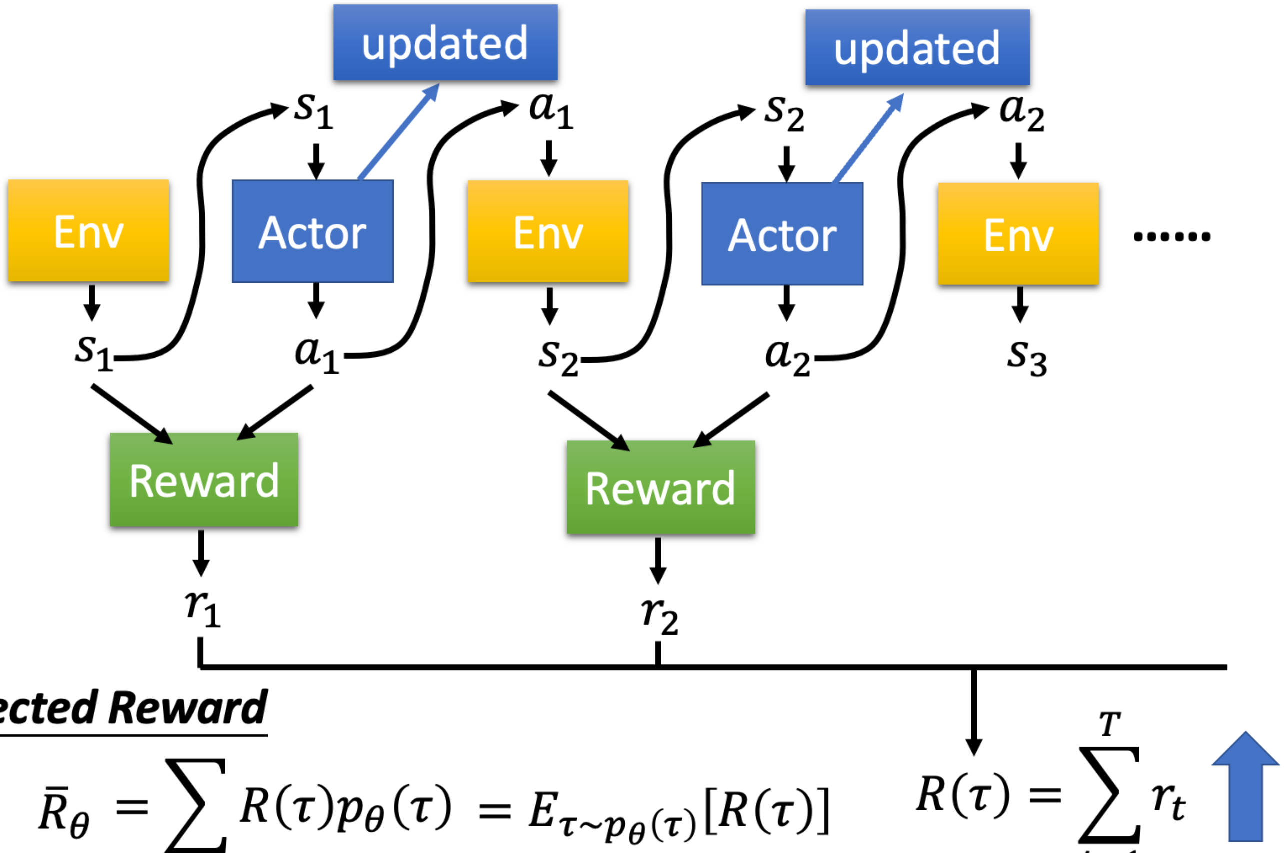
$$R(\tau) = \sum_{t=1}^{T} r_t$$

Reinforcement Learning vs. Text Generation





Reinforcement Learning



Expected Reward

$$\bar{R}_\theta = \sum_\tau R(\tau) p_\theta(\tau) = E_{\tau \sim p_\theta(\tau)}[R(\tau)]$$

$$R(\tau) = \sum_{t=1}^{T} r_t$$

Solutions

- Q-Learning
- Policy Gradient
- Actor-Critic
- …

Optimizing for Human Preferences

How do we change the LM parameters θ to maximize this?

$$\mathbb{E}_{\hat{s}\sim p_\theta(s)}[R(\hat{s})]$$

Gradient Ascent

$$\theta_{t+1} \coloneqq \theta_t + \alpha \nabla_{\theta_t} \mathbb{E}_{\hat{s}\sim p_{\theta_t}(s)}[R(\hat{s})]$$

Policy Gradient Methods in Reinforcement Learning
(REINFORCE) [Williams, 1992]

Policy Gradient/REINFORCE

Gradient Ascent

$$\theta_{t+1} \coloneqq \theta_t + \alpha \, \nabla_{\theta_t} \mathbb{E}_{\hat{s} \sim p_{\theta_t}(s)}[R(\hat{s})]$$

$$\nabla_\theta \mathbb{E}_{\hat{s} \sim p_\theta(s)}[R(\hat{s})] = \nabla_\theta \sum_s R(s) p_\theta(s) = \sum_s R(s) \, \nabla_\theta p_\theta(s)$$

Log-Derivative Trick

$$\nabla_\theta \log p_\theta(s) = \frac{1}{p_\theta(s)} \nabla_\theta p_\theta(s) \qquad \Longrightarrow \qquad \nabla_\theta \, p_\theta(s) = \nabla_\theta \log p_\theta(s) \, p_\theta(s)$$

Policy Gradient/REINFORCE

$$\nabla_\theta \mathbb{E}_{\hat{s}\sim p_\theta(s)}[R(\hat{s})] = \sum_s R(s)\, \nabla_\theta p_\theta(s) = \sum_s p_\theta(s) R(s)\, \nabla_\theta \log p_\theta(s)$$

$$= \mathbb{E}_{\hat{s}\sim p_\theta(s)}[R(\hat{s})\, \nabla_\theta \log p_\theta(\hat{s})]$$

We can approximate this objective with Monte Carlo samples

$$\nabla_\theta \mathbb{E}_{\hat{s}\sim p_\theta(s)}[R(\hat{s})] = \mathbb{E}_{\hat{s}\sim p_\theta(s)}[R(\hat{s})\, \nabla_\theta \log p_\theta(\hat{s})] \approx \frac{1}{m}\sum_{i=1}^{m} R(s_i)\, \nabla_\theta \log p_\theta(s_i)$$

Policy Gradient/REINFORCE

If R is +++

Take gradient steps to maximize $p_\theta(s_i)$

$$\theta_{t+1} \coloneqq \theta_t + \alpha \frac{1}{m} \sum_{i=1}^{m} R(s_i)\, \nabla_{\theta_t} \log p_{\theta_t}(s_i)$$

If R is ---

Take steps to minimize $p_\theta(s_i)$

We reinforce good actions, increasing the chance they happen again

Proximal Policy Optimization (PPO)

- New parameters θ' cannot be very different from old parameters θ

$$J_{PPO}^{\theta'}(\theta) = J^{\theta'}(\theta) - \beta KL(\theta, \theta')$$

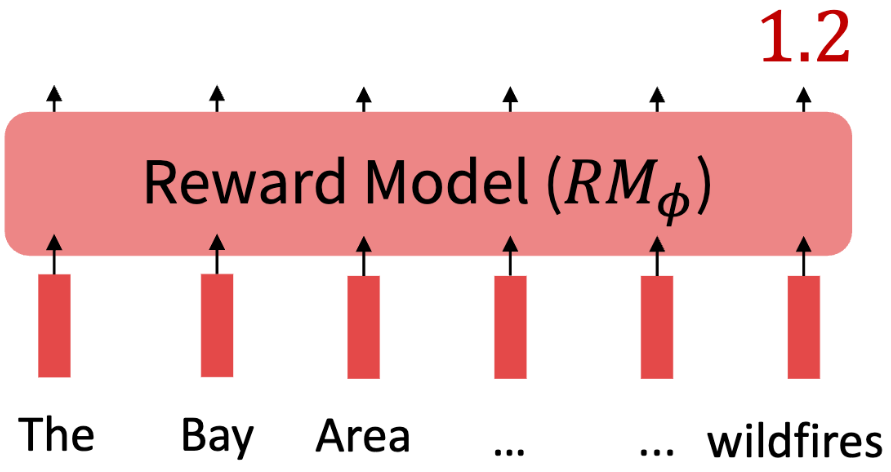
Regularization

Human Feedback In The Training Loop

- Training loop
  - The model generates a sentence
  - A human expert decides the reward
  - Update the model by policy gradient
- Should we always use real human feedback?
  - Human experts are expensive
  - Training process is not automated

How to Model Human Preferences?

- Now for any reward function R, we can train our language model to maximize expected reward
- Problem 1: human-in-the-loop is expensive
  - Solution: instead of directly asking humans for preferences, model their preferences as a separate (NLP) problem
  - Train a reward model (RM) from an annotated dataset

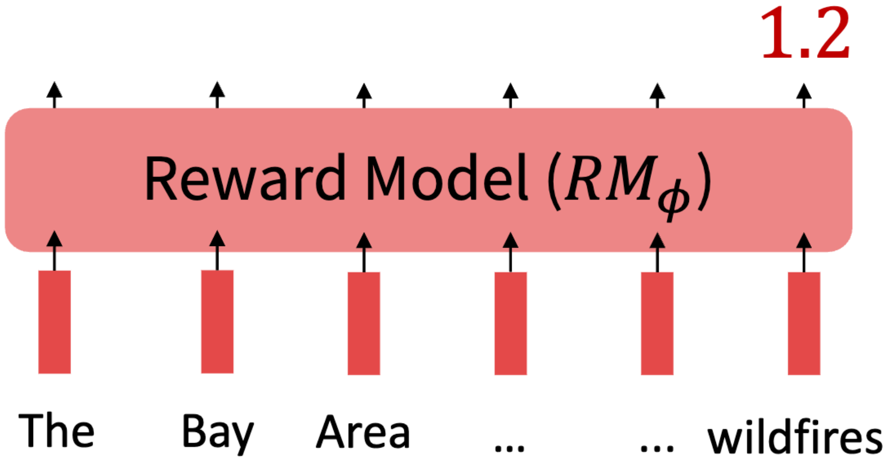


How to Model Human Preferences?

- Now for any reward function R, we can train our language model to maximize expected reward
- Problem 2: human judgments are noisy and miscalibrated
  - Solution: instead of asking for direct ratings, ask for pairwise comparisons, which can be more reliable

An earthquake hit San Francisco. There was minor property damage, but no injuries.	>	A 4.2 magnitude earthquake hit San Francisco, resulting in massive damage.	>	The Bay Area has good weather but is prone to earthquakes and wildfires.
s_1		s_3		s_2

Training A Reward Model



Bradley-Terry [1952] paired comparison model

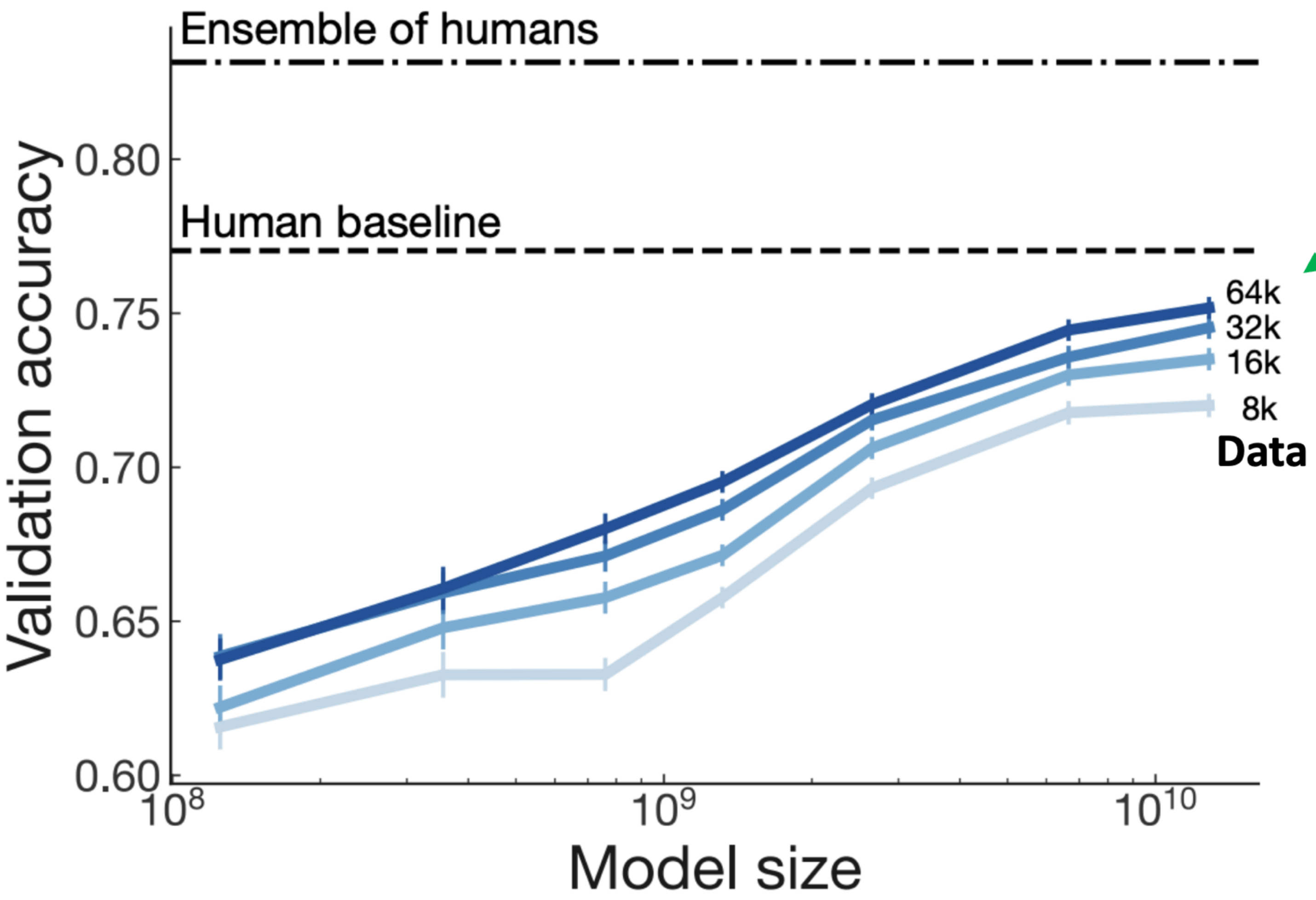
$$J_{RM}(\phi) = -\mathbb{E}_{(s^w, s^l) \sim D}\left[\log \sigma(RM_\phi(s^w) - RM_\phi(s^l))\right]$$

"winning" sample

"losing" sample

s^w should score higher than s^l

Reward Model vs. Real Human Feedback



Large enough RM trained on enough data approaching single human perf

[Stiennon et al., 2020]

RLHF: Putting Everything All Together

- We have the following:
  - A pretrained (possibly instruction-finetuned) LM $p^{PT}(y \mid x)$
  - A reward model $RM_{\phi}(x, y)$ that produces scalar rewards for LM outputs, trained on a dataset of human comparisons
- Now to do RLHF:
  - Copy the model $p_{\theta}^{RL}(y \mid x)$, with parameters θ we would like to optimize
  - We want to optimize:

$$\mathbb{E}_{\hat{y} \sim p_{\theta}^{RL}(\hat{y} \mid x)} \left[RM_{\phi}(x, \hat{y}) \right]$$

RLHF: Putting Everything All Together

- We want to optimize:

$$\mathbb{E}_{\hat{y} \sim p_\theta^{RL}(\hat{y}|x)} \left[RM_\phi(x, \hat{y}) \right]$$

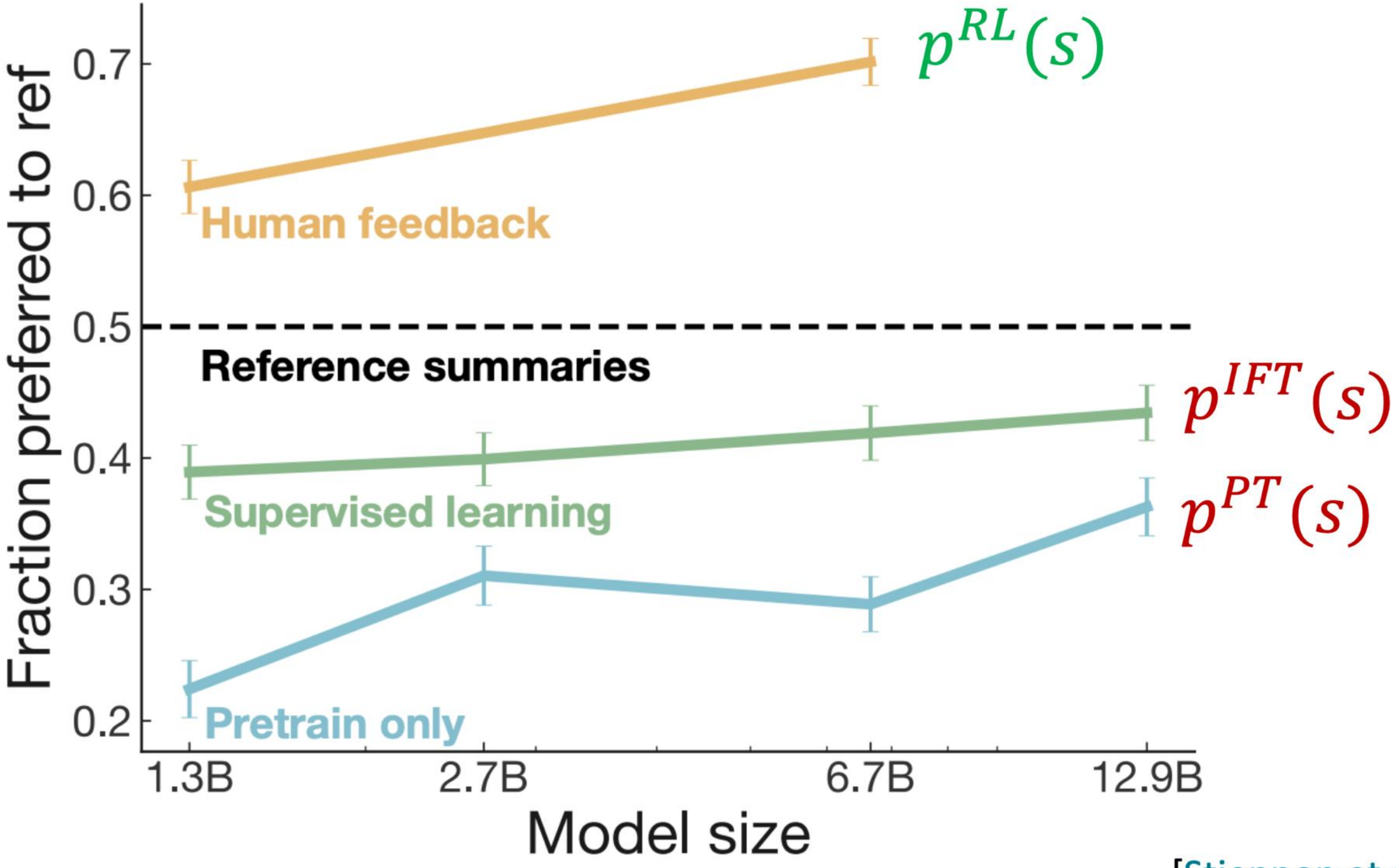
- Do you see any problems?
  - Learned rewards are imperfect; this quantity can be imperfectly optimized
- Add a penalty for drifting too for from the initialization:

$$\mathbb{E}_{\hat{y} \sim p_\theta^{RL}(\hat{y}|x)} \left[RM_\phi(x, \hat{y}) \ - \beta \underbrace{\log \left(\frac{p_\theta^{RL}(\hat{y} \mid x)}{p^{PT}(\hat{y} \mid x)} \right)}] \right.$$

Pay a price when
$p_\theta^{RL}(\hat{y} \mid x) > p^{PT}(\hat{y} \mid x)$

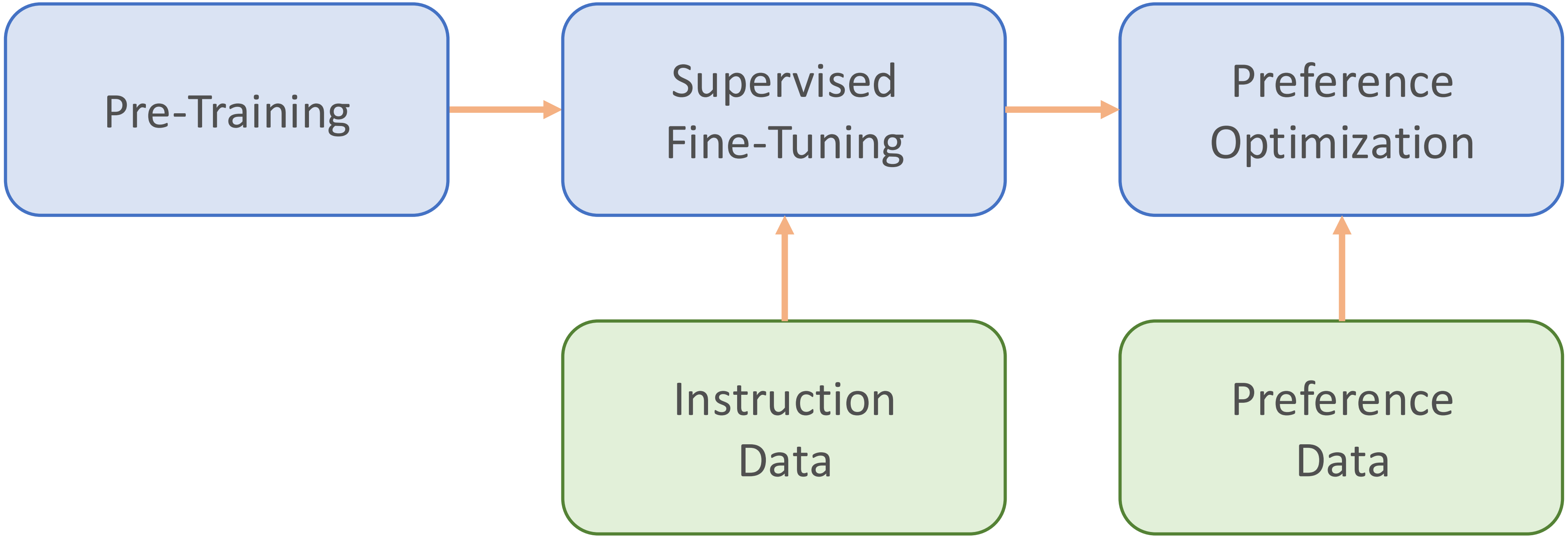
This penalty which prevents us from diverging too far from the pretrained model. In expectation, it is known as the **Kullback-Leibler** (**KL**) divergence between $p_\theta^{RL}(\hat{y} \mid x)$ and $p^{PT}(\hat{y} \mid x)$.

RLHF vs. Supervised Fine-Tuning

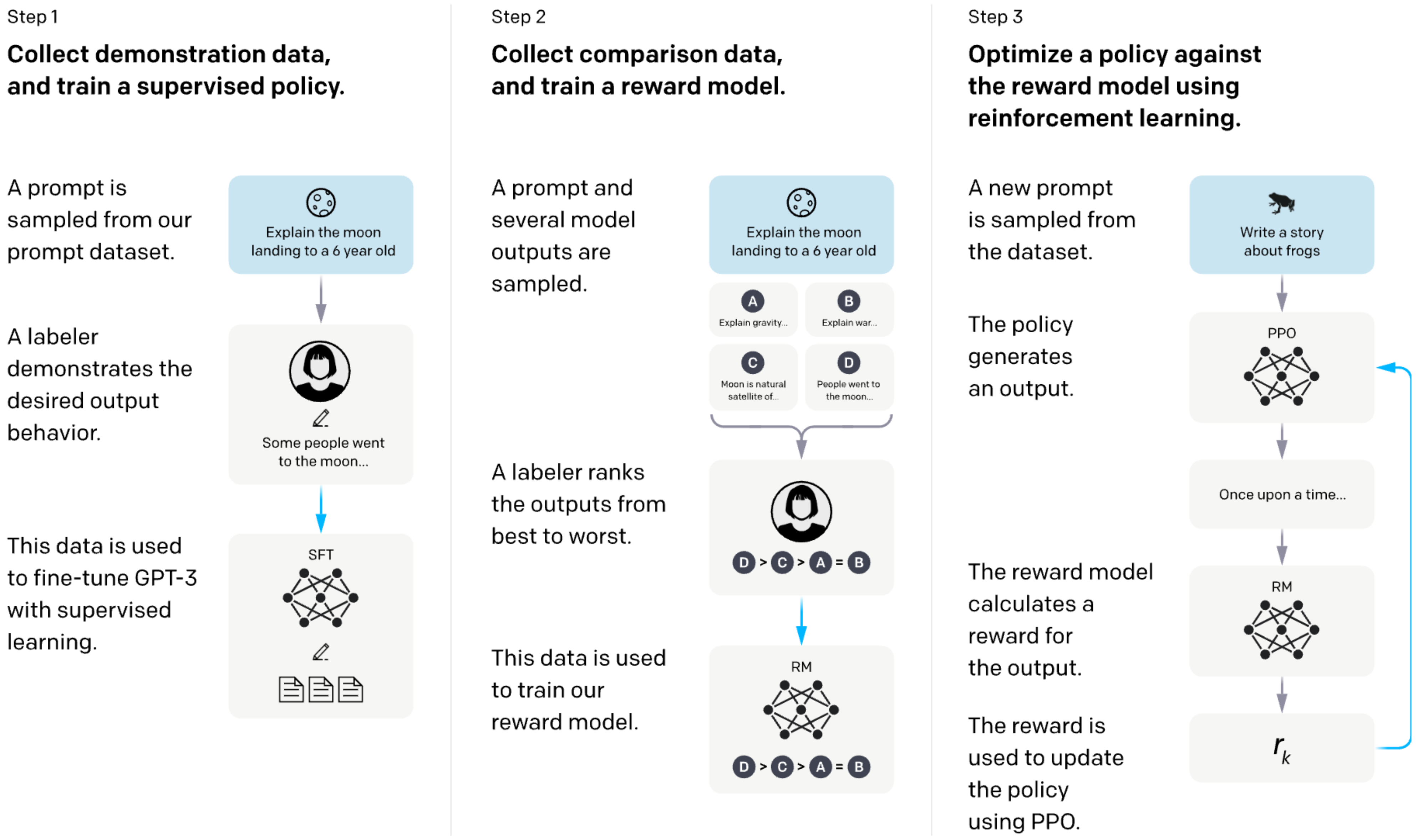


[Stiennon et al., 2020]

Post-Training Pipeline



InstructGPT



Step 1

Collect demonstration data, and train a supervised policy.

A prompt is sampled from our prompt dataset.

A labeler demonstrates the desired output behavior.

This data is used to fine-tune GPT-3 with supervised learning.

Explain the moon landing to a 6 year old
Some people went to the moon...
SFT

Step 2

Collect comparison data, and train a reward model.

A prompt and several model outputs are sampled.

A labeler ranks the outputs from best to worst.

This data is used to train our reward model.

Explain the moon landing to a 6 year old
A
Explain gravity...
B
Explain war...
C
Moon is natural satellite of...
D
People went to the moon...
D > C > A = B
RM
D > C > A = B

Step 3

Optimize a policy against the reward model using reinforcement learning.

A new prompt is sampled from the dataset.

The policy generates an output.

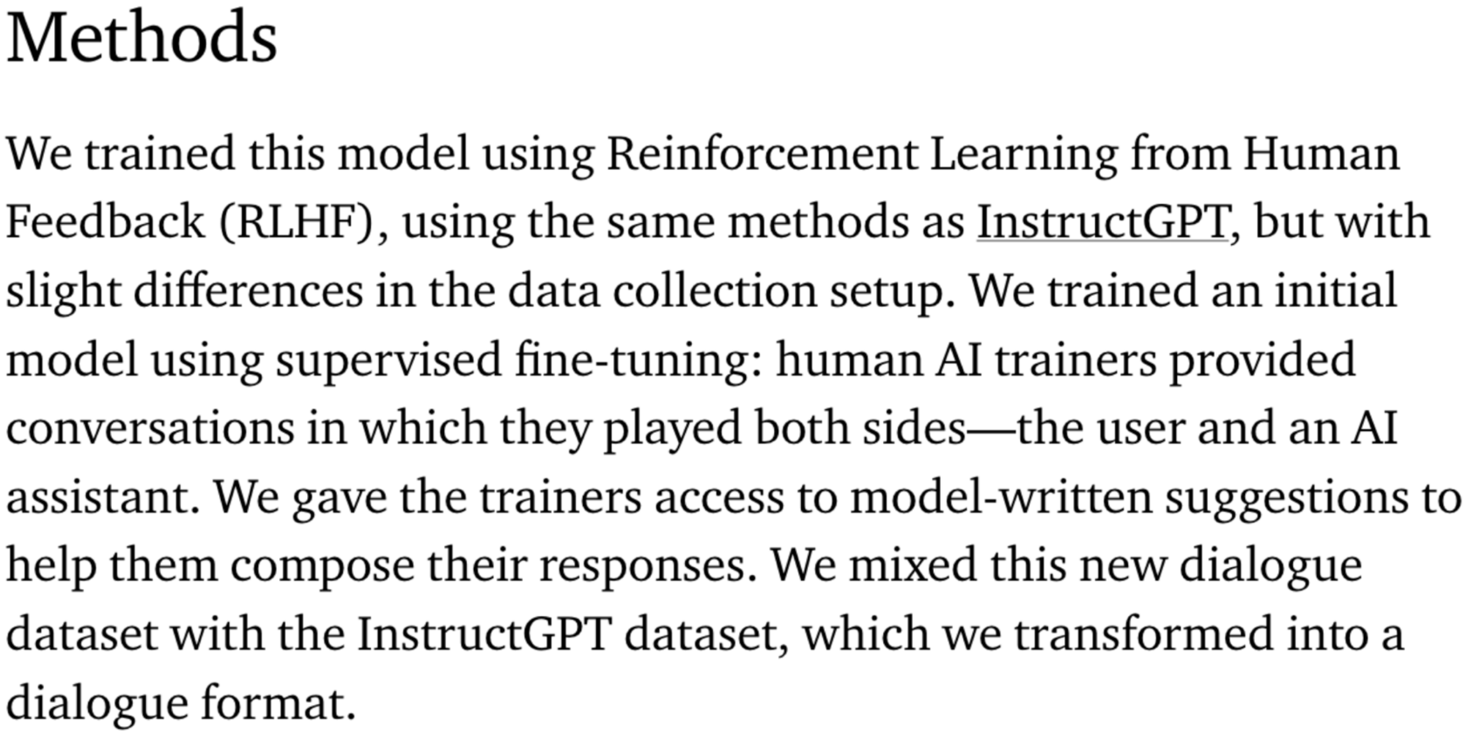
The reward model calculates a reward for the output.

The reward is used to update the policy using PPO.

ChatGPT: Instruction Fine-tuning + RLHF for Dialog Agents

ChatGPT: Optimizing Language Models for Dialogue

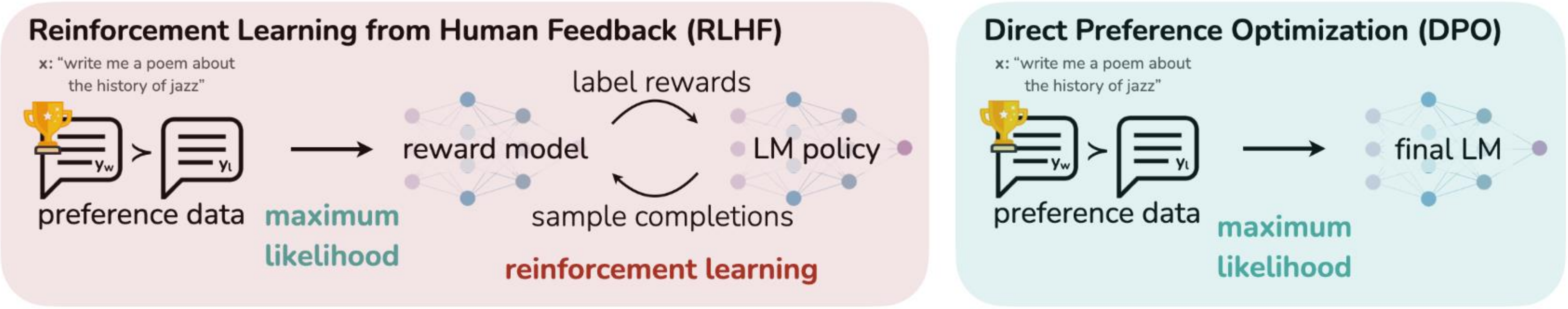
Note: OpenAI (and similar companies) are keeping more details secret about ChatGPT training (including data, training parameters, model size)—perhaps to keep a competitive edge…



Methods

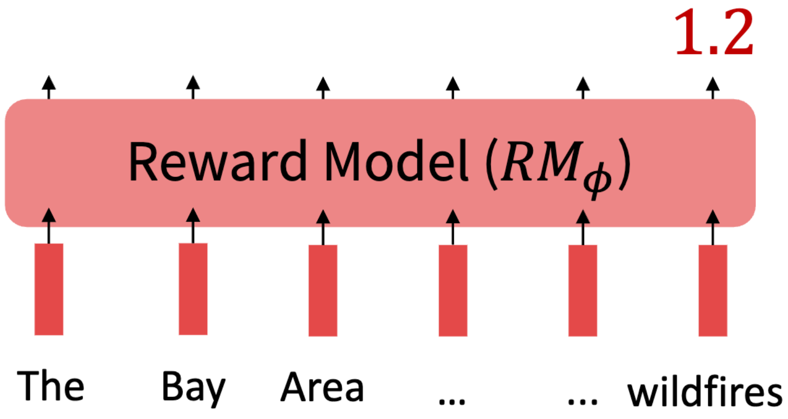
We trained this model using Reinforcement Learning from Human Feedback (RLHF), using the same methods as InstructGPT, but with slight differences in the data collection setup. We trained an initial model using supervised fine-tuning: human AI trainers provided conversations in which they played both sides—the user and an AI assistant. We gave the trainers access to model-written suggestions to help them compose their responses. We mixed this new dialogue dataset with the InstructGPT dataset, which we transformed into a dialogue format.

Direct Preference Optimization (DPO)


Reinforcement Learning from Human Feedback (RLHF)
x: "write me a poem about the history of jazz"
y_w
y_l
preference data
maximum likelihood
reward model
label rewards
sample completions
LM policy
reinforcement learning
Direct Preference Optimization (DPO)
x: "write me a poem about the history of jazz"
y_w
y_l
preference data
maximum likelihood
final LM

Direct Preference Optimization: Your Language Model is Secretly a Reward Model, 2023

RLHF: Proximal Policy Optimization (PPO)



An earthquake hit San Francisco. There was minor property damage, but no injuries.

$>$

The Bay Area has good weather but is prone to earthquakes and wildfires.

s_1 s_2

$$\mathcal{L}_R(r_\phi, \mathcal{D}) = -\mathbb{E}_{(x,y_w,y_l)\sim\mathcal{D}}\left[\log\sigma(r_\phi(x, y_w) - r_\phi(x, y_l))\right]$$

Direct Preference Optimization (DPO)

RLHF Objective

(get **high reward**, stay **close** to reference model)

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

Maximize reward

Keep similar behavior

$$
\begin{aligned}
&\max_{\pi} \mathbb{E}_{x\sim\mathcal{D}, y\sim\pi} \left[r(x,y)\right] - \beta \mathbb{D}_{\mathrm{KL}}\left[\pi(y|x) \;||\; \pi_{\mathrm{ref}}(y|x)\right] \\
&= \max_{\pi} \mathbb{E}_{x\sim\mathcal{D}} \mathbb{E}_{y\sim\pi(y|x)} \left[r(x,y) - \beta \log \frac{\pi(y|x)}{\pi_{\mathrm{ref}}(y|x)}\right] \\
&= \min_{\pi} \mathbb{E}_{x\sim\mathcal{D}} \mathbb{E}_{y\sim\pi(y|x)} \left[\log \frac{\pi(y|x)}{\pi_{\mathrm{ref}}(y|x)} - \frac{1}{\beta} r(x,y)\right] \\
&= \min_{\pi} \mathbb{E}_{x\sim\mathcal{D}} \mathbb{E}_{y\sim\pi(y|x)} \left[\log \frac{\pi(y|x)}{\frac{1}{Z(x)} \pi_{\mathrm{ref}}(y|x) \exp\left(\frac{1}{\beta} r(x,y)\right)} - \log Z(x)\right]
\end{aligned}
$$

$$Z(x) = \sum_{y} \pi_{\mathrm{ref}}(y|x) \exp\left(\frac{1}{\beta} r(x,y)\right)$$

Direct Preference Optimization (DPO)

RLHF Objective

(get **high reward**, stay **close** to reference model)

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

Maximize reward

Keep similar behavior

$$\pi^*(y|x) = \frac{1}{Z(x)}\pi_{\mathrm{ref}}(y|x)\exp\left(\frac{1}{\beta}r(x, y)\right)$$

$$\min_{\pi} \mathbb{E}_{x\sim\mathcal{D}}\mathbb{E}_{y\sim\pi(y|x)}\left[\log\frac{\pi(y|x)}{\frac{1}{Z(x)}\pi_{\mathrm{ref}}(y|x)\exp\left(\frac{1}{\beta}r(x, y)\right)} - \log Z(x)\right]$$

$$= \min_{\pi} \mathbb{E}_{x\sim\mathcal{D}}\left[\mathbb{E}_{y\sim\pi(y|x)}\left[\log\frac{\pi(y|x)}{\pi^*(y|x)}\right] - \log Z(x)\right]$$

$$= \min_{\pi} \mathbb{E}_{x\sim\mathcal{D}}\left[\mathbb{D}_{\mathrm{KL}}(\pi(y|x) \mid\mid \pi^*(y|x)) - \log Z(x)\right]$$

$$\pi(y|x) = \pi^*(y|x) = \frac{1}{Z(x)}\pi_{\mathrm{ref}}(y|x)\exp\left(\frac{1}{\beta}r(x, y)\right)$$

Direct Preference Optimization (DPO)

RLHF Objective

(get **high reward**, stay **close** to reference model)

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

Maximize reward

Keep similar behavior

Closed-form Optimal Policy

(write **optimal policy** as function of **reward function**; from prior work)

$$\pi^*(y \mid x) = \frac{1}{Z(x)}\pi_{\mathrm{ref}}(y \mid x)\exp\left(\frac{1}{\beta}r(x,y)\right)$$

with $Z(x) = \sum_{y} \pi_{\mathrm{ref}}(y \mid x)\exp\left(\frac{1}{\beta}r(x,y)\right)$

Note **intractable sum** over possible responses; can't immediately use this

Rearrange

(write **any reward function** as function of **optimal policy**)

Ratio is **positive** if policy likes response more than reference model, **negative** if policy likes response less than ref. model

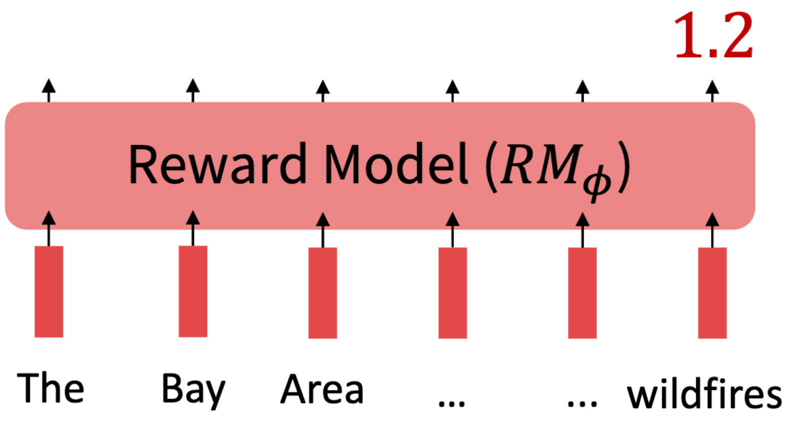
$$r(x,y) = \underbrace{\beta\log\frac{\pi^*(y \mid x)}{\pi_{\mathrm{ref}}(y \mid x)} + \beta\log Z(x)}_{\text{some parameterization of a reward function}}$$

Direct Preference Optimization (DPO)

A loss function on <u>reward functions</u>

Derived from the Bradley-Terry model of human preferences:

$$\mathcal{L}_R(r, \mathcal{D}) = -\mathbb{E}_{(x, y_w, y_l)\sim\mathcal{D}}\left[\log \sigma(r(x, y_w) - r(x, y_l))\right]$$



```
An earthquake hit
San Francisco.
There was minor
property damage,
but no injuries.
```

s_1

$>$

```
The Bay Area has
good weather but is
prone to
earthquakes and
wildfires.
```

s_2

Direct Preference Optimization (DPO)

A loss function on reward functions

Derived from the Bradley-Terry model of human preferences:

$$\mathcal{L}_R(r, \mathcal{D}) = -\mathbb{E}_{(x, y_w, y_l) \sim \mathcal{D}} \left[\log \sigma(r(x, y_w) - r(x, y_l)) \right]$$

+

A transformation between reward functions and policies

$$r_{\pi_\theta}(x, y) = \beta \log \frac{\pi_\theta(y \mid x)}{\pi_{\text{ref}}(y \mid x)} + \beta \log Z(x)$$

Direct Preference Optimization (DPO)

A loss function on <u>reward functions</u>

Derived from the Bradley-Terry model of human preferences:

$$\mathcal{L}_R(r, \mathcal{D}) = -\mathbb{E}_{(x, y_w, y_l) \sim \mathcal{D}} \left[\log \sigma(r(x, y_w) - r(x, y_l)) \right]$$

+

A transformation between <u>reward functions</u> and <u>policies</u>

$$r_{\pi_\theta}(x, y) = \beta \log \frac{\pi_\theta(y \mid x)}{\pi_{\text{ref}}(y \mid x)} + \beta \log Z(x)$$

=

A loss function on <u>policies</u>

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

Direct Preference Optimization (DPO)

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

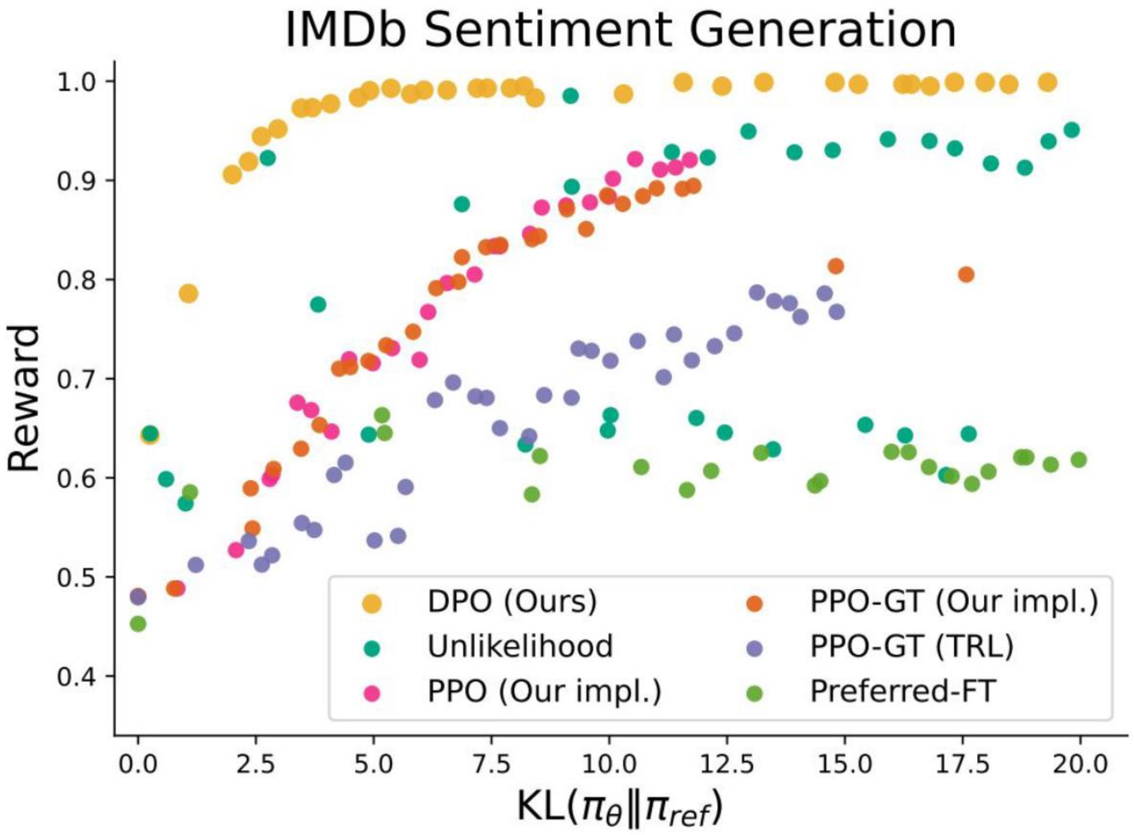
Reward of **preferred** response

Reward of **dispreferred** response

$$\nabla_\theta \mathcal{L}_{\text{DPO}}(\pi_\theta; \pi_{\text{ref}}) =$$

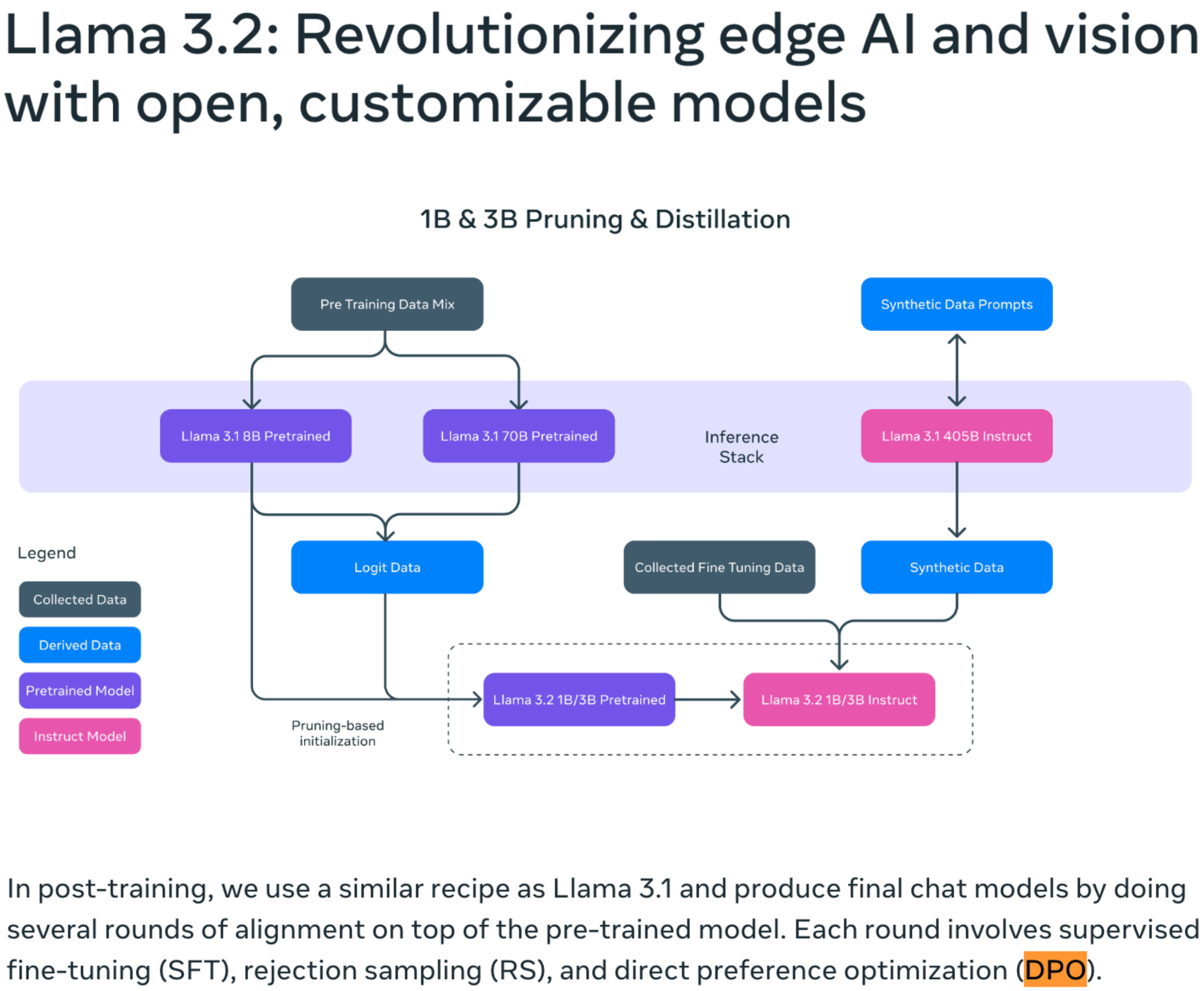
$$-\beta \mathbb{E}_{(x, y_w, y_l) \sim \mathcal{D}} \bigg[\underbrace{\sigma(\hat{r}_\theta(x, y_l) - \hat{r}_\theta(x, y_w))}_{\text{higher weight when reward estimate is wrong}} \bigg[\underbrace{\nabla_\theta \log \pi(y_w \mid x)}_{\text{increase likelihood of } y_w} - \underbrace{\nabla_\theta \log \pi(y_l \mid x)}_{\text{decrease likelihood of } y_l} \bigg] \bigg]$$

DPO Performance



1. Generate positive IMDB reviews from GPT2-XL
2. Use pre-trained sentiment classifier as Gold RM
3. Create preferences based on Gold RM
4. Optimize with PPO and DPO

Large-Scale DPO Training



Llama 3.2: Revolutionizing edge AI and vision with open, customizable models

In post-training, we use a similar recipe as Llama 3.1 and produce final chat models by doing several rounds of alignment on top of the pre-trained model. Each round involves supervised fine-tuning (SFT), rejection sampling (RS), and direct preference optimization (DPO).

Lecture Plan

- Post-Training
  - Alignment
  - Instruction Tuning
  - RLHF/PPO
  - DPO