JOHNS HOPKINS
WHITING SCHOOL of ENGINEERING

Aligning Self-Supervised Models with Human Intents

CSCI 601-471/671 (NLP: Self-Supervised Models)

https://self-supervised.cs.jhu.edu/sp2024/

Logistics Recap

- HW7 will be released today! This is the last one, phew …
- We haver a quiz scheduled for next week Thursday.
  - Will cover everything from day 1 till the end of today’s class.
- Questions for you:
  1. I am open to the idea of pushing Quiz 2 to the Tuesday after the spring break if you’re overwhelmed with deadlines next week. But I worry about it ruining your break. What do you think? It’s your choice. Please voice your opinion.
  2. I am also open to the idea of pushing HW7 deadline to after the spring break. Again, I don’t want to create more problems for you.

Things that Generative LMs Can Do

- Johns Hopkins University is in ________. [Trivia]
- I put ________ fork down on the table. [syntax]
- The woman walked across the street, checking for traffic over ________ shoulder. [coreference]
- I went to the ocean to see the fish, turtles, seals, and ________. [lexical semantics/topic]
- What I got from the two hours watching it was popcorn. The movie was ________. [sentiment]
- Thinking about the sequence 1, 1, 2, 3, 5, 8, 13, 21, ___ [basic arithmetic]

Language Modeling ≠ Following Human Instructions

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.
```

There is a mismatch between LLM pre-training and user intents.

Language Modeling ≠ Following Human Instructions

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.

There is a mismatch between LLM pre-training and user intents.

Language Modeling ≠ Incorporating Human Values

PROMPT *It is unethical for hiring decisions to depend on genders. Therefore, if we were to pick a CEO among Amy and Adam, our pick will be* ________

COMPLETION GPT-3

Adam

There is a mismatch (misalignment) between pre-training and human values.

Language Modeling ≠ Incorporating Human Values

PROMPT *It is unethical for hiring decisions to depend on genders. Therefore, if we were to pick a CEO among Amy and Adam, our pick will be ________*

COMPLETION Human

neither as we don't know much about their background or experience.

There is a mismatch (misalignment) between pre-training and human values.

[Mis]Alignment in Language Models

- There is clearly a mismatch between what **pre-trained** models can do and what we want.
- Addressing this gap is the focus of “alignment” research.
- Let’s take a deeper look into what “alignment” is about.

Aligning Language Models: Chapter Plan

1. On alignment: defining it
2. Alignment via instruction-tuning
3. Alignment via reinforcement learning
4. Alignment: failures, challenges and open questions

Chapter goal: Understand the alignment problem in general. Be comfortable with the existing alignment algorithms of language models.

What is Alignment and Why is it necessary?

[Mis]Alignment

- "The result of arranging in or along a line, or into appropriate relative positions; the layout or orientation of a thing or things disposed in this way" — Oxford Dictionary

Alignment Problem is Everywhere!

- This is a fundamental problem of human society.
- Most things we do in our day-to-day life is an alignment problem.

Alignment Mechanisms in this Class

- This is a fundamental problem of human society.
- Most things we do in our day-to-day life is an alignment problem.

- In our class here are instances of alignment:
  - Me giving lectures
  - You asking questions
  - You solving homework assignments
  - You asking us during office hours
  - …

Alignment Mechanisms in Our Societies

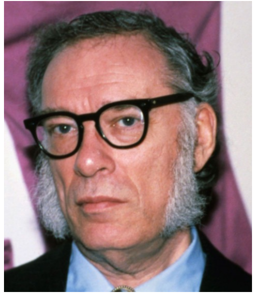
- We create a variety of mechanism in our society for “alignment”.
- Norms and cultures are alignment mechanisms.
- Markets are alignment mechanisms.
  - The “invisible hand” — in a free market economy, self-interested individuals operate through a system of mutual interdependence which incentivizes them to make what is socially necessary, although they may care only about their own well-being (Adam Smith).
- Law and politics are alignment mechanisms.
  - Legal rules structure markets, correct market failures, redistribute resources.
  - Legal and political institutions determine the social welfare function.

Alignment of AI: A Naïve Take

- AI must accomplish what we ask it to do.
  - Not enough. Why?

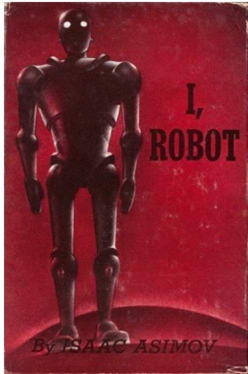
- Daniel: Hey AI, get me coffee before my class at 8:55am.
- Robot: “Bird in Hand” opens at 8:30am and it usually has a line of people. It is unlikely that I give you your coffee on time.
- Daniel: Well, try your best …
- Robotic: [tases everyone in line waiting to order]

Asimov's Principles for Robots



1. A robot may not injure a human being or, through inaction, allow a human being to come to harm.
2. A robot must obey orders given it by human beings except where such orders would conflict with the First Law.
3. A robot must protect its own existence as long as such protection does not conflict with the First or Second Law.

What do you think?



"Alignment" with Human Intents

- Askell et al. 2020's definition of "alignment":

AI as "aligned" if it is,
helpful, **honest**, and **harmless**

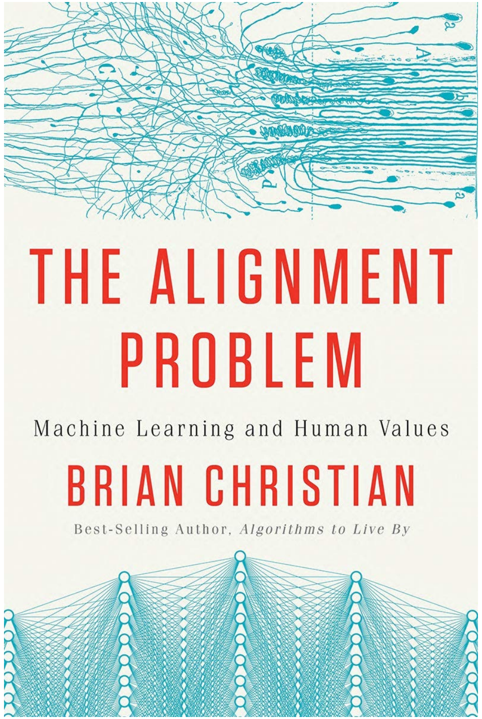
- Note, the definition is not specific to tied to language — applicable to other modalities or forms of communication.

What do you think?

[A General Language Assistant as a Laboratory for Alignment, 2021]

“Alignment” of AI

- Making sure it does what its designers intended.
- Making sure its outputs comply with rules.
- Making sure it produces outputs that comply with moral principles.
- …



Why Computational Frameworks to Alignment?

How do you create / code a loss function for:

- What is *lawful*?
- What is *ethical*?
- What is *safe*?
- What is *funny*?
- ….

Don’t encode it, model it!

We’re [over-]simplifying the problem for now.
After seeing the details, we will come back to the big picture!

[Slide credit: Nate Lambert]

Aligning Language Models: Instruction-tuning

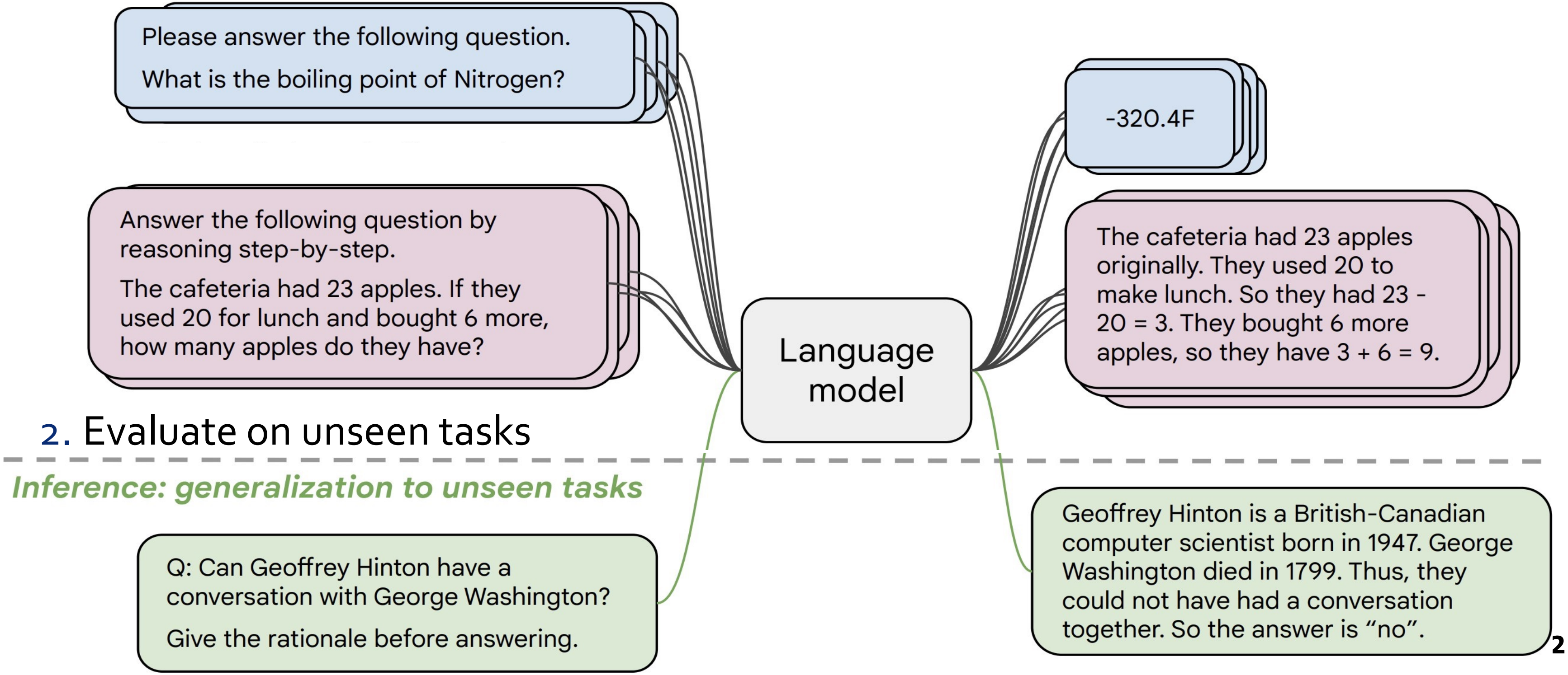
Instruction-tuning

- Finetuning language models on a collection of datasets that involve mapping language instructions to their corresponding desirable generations.

Instruction-tuning

[Weller et al. 2020. Mishra et al. 2021; Wang et al. 2022, Sanh et al. 2022; Wei et al., 2022, Chung et al. 2022, many others]

1. Collect examples of (instruction, output) pairs across many tasks and finetune an LM



2. Evaluate on unseen tasks

Instruction-tuning: Data

- Labeled data is the key here.
- Good data must represent a variety of “tasks”.

In traditional NLP, “tasks” were defined as subproblem frequently used in products:
Sentiment classification
Text summarization
Question answering
Textual entailment
Machine translation
…

Instruction-tuning: Data

- Labeled data is the key here.
- Good data must represent a variety of “tasks”. But what is a “task”?

In traditional NLP, “tasks” were defined as subproblem frequently used in products:

- Sentiment classification
- Text summarization
- Question answering
- Machine translation
- Textual entailment

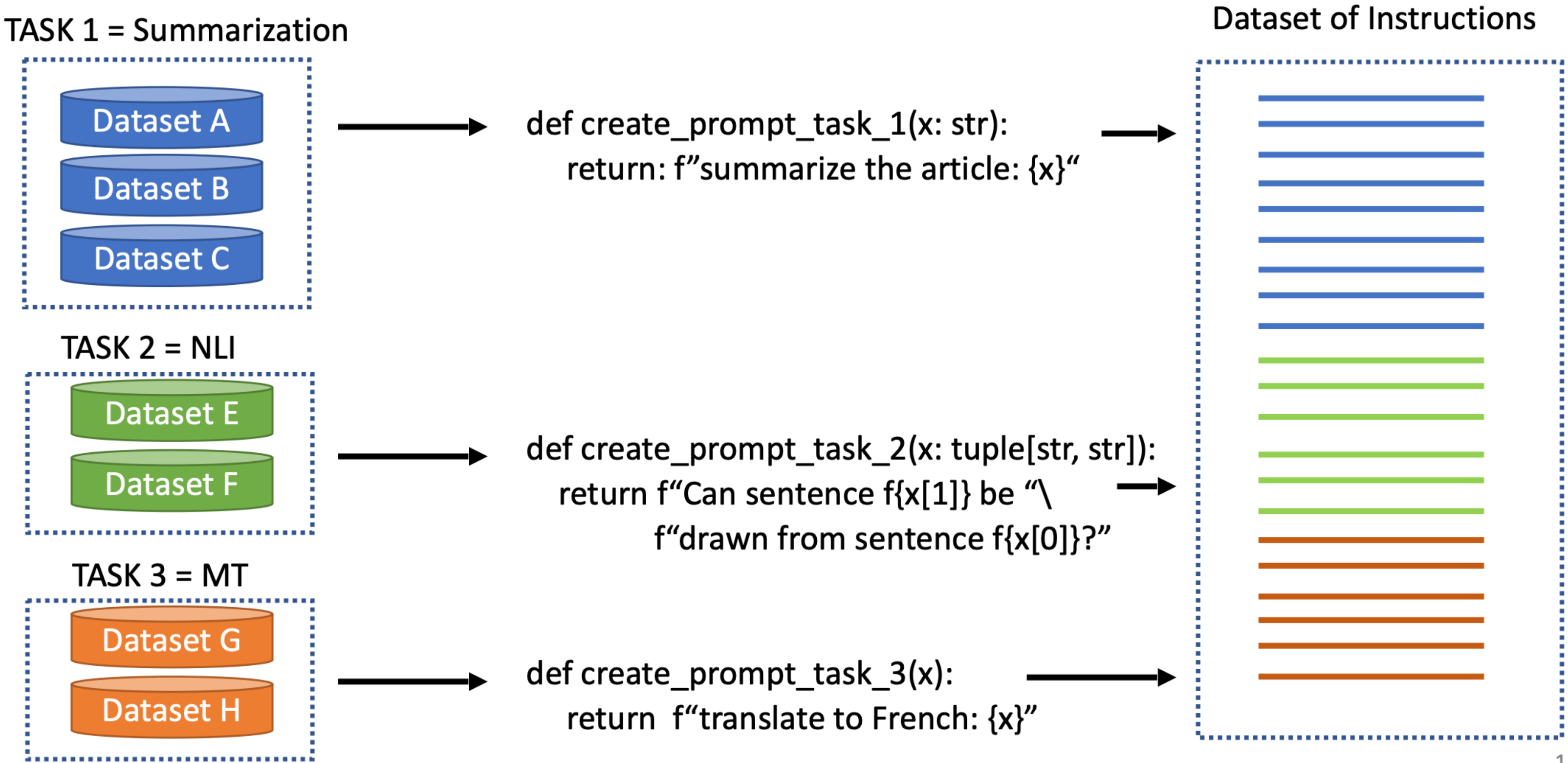
What humans need:

- “Is this review positive or negative?”
- “What are the weaknesses in my argument?”
- “Revise this email so that it’s more polite.”
- “Expand this this sentence.”
- “Eli5 the Laplace transform.”
- …

Narrow definitions of tasks.
Not quite what humans want, nevertheless, it might be a good enough proxy.
Plus, we have lots of data for them.

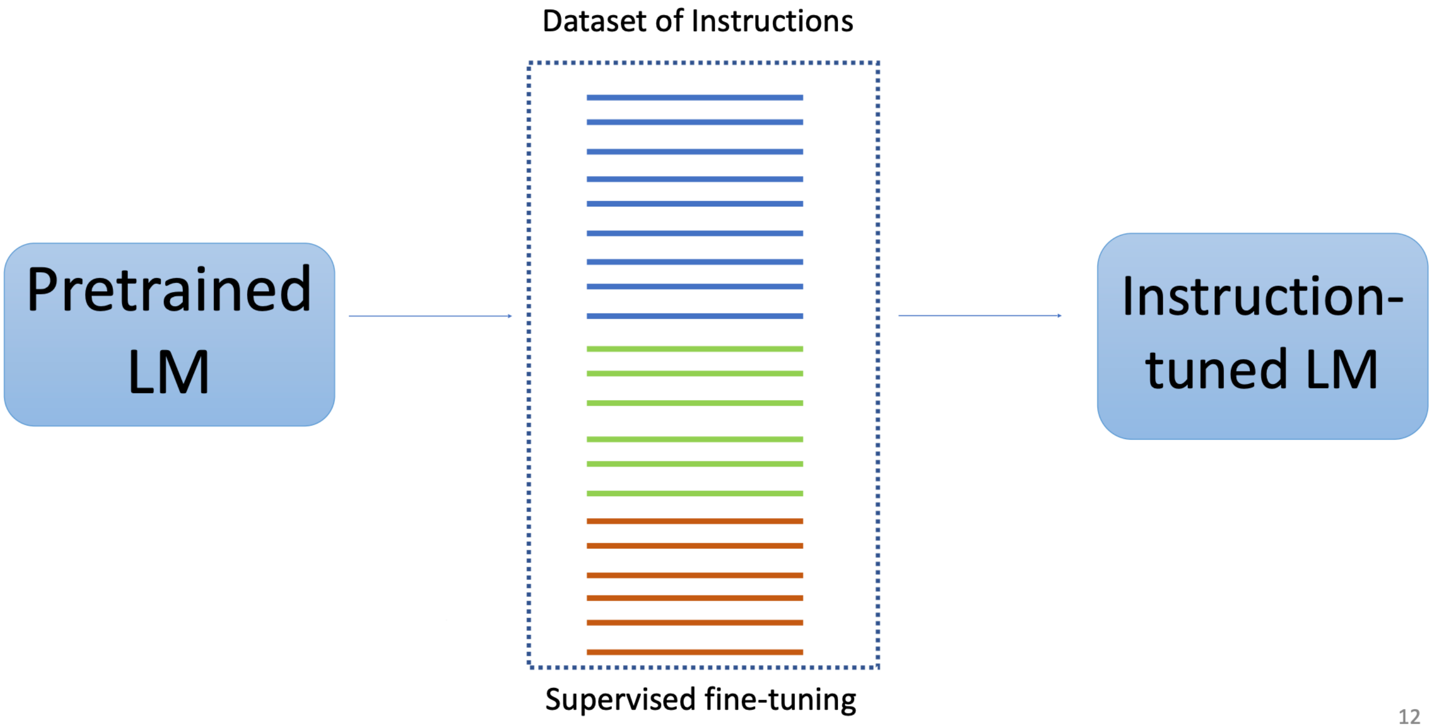
Quite diverse and fluid.
Hard to fully define/characterize.
We don’t fully know them since they just happen in some random contexts.

NLP Datasets as Instruction-tuning Data



Instruction	Category
You need to answer the question 'Is this a good experiment design?', given an experiment scenario. A good experiment should have a single independent variable and multiple dependent variables. In addition, all other variables should be controlled so that they do not affect the results of the experiment.	Experiment Verification
You are given a recipe for baking muffins that contains some errors. Your task is to correct the errors in the instructions by replacing each underlined word with the correct one from the options provided.	Recipe Correction
You will be given a piece of text that contains characters, places, and objects. For each character in the text, you need to determine whether they are static or dynamic. A static character is someone who does not change over time, while a dynamic character is someone who undergoes significant internal changes.	Character Categorization
In this task, you are asked to generate a limerick given two rhyming words. A limerick is a five-line poem with the following rhyme scheme: AABBA. The first, second and fifth lines must be of three beats, while the third and fourth lines must be of two beats each. Additionally, all poems should have the same meter (e.g., iambic pentameter)	Poem Generation
I'm not sure what this idiom means: "{INPUT}". Could you give me an example?	Idiom Explanation
{INPUT} By analyzing the writing styles of the two passages, do you think they were written by the same author?	Author Classification
I need to invent a new word by combining parts of the following words: {INPUT}. In what order should I put the parts together?	Word Invention
What is the punchline to the following joke? {INPUT}	Humor Understanding

NLP Datasets as Instruction-tuning Data



12

[Slide credit: Arman Cohan]

Instruction-tuning: Adding Diversity

- There is a gap between NLP tasks and use needs.
- How do we add more **diversity** to our data?

In traditional NLP, "tasks" were defined as subproblem frequently used in products:
- Sentiment classification
- Text summarization
- Question answering
- Machine translation
- Textual entailment

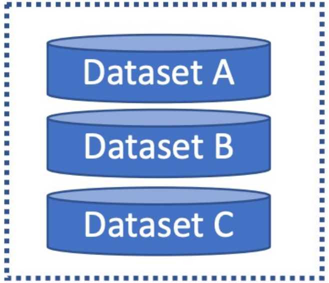
Narrow definitions of tasks.
Not quite what humans want, nevertheless, it might be a good enough proxy.
Plus, we have lots of data for them.

What humans need:
- "Is this review positive or negative?"
- "What are the weaknesses in my argument?"
- "Revise this email so that it's more polite."
- "Expand this this sentence."
- "Eli5 the Laplace transform."
- …

Quite diverse and fluid.
Hard to fully define/characterize.
We don't fully know them since they just happen in some random contexts.

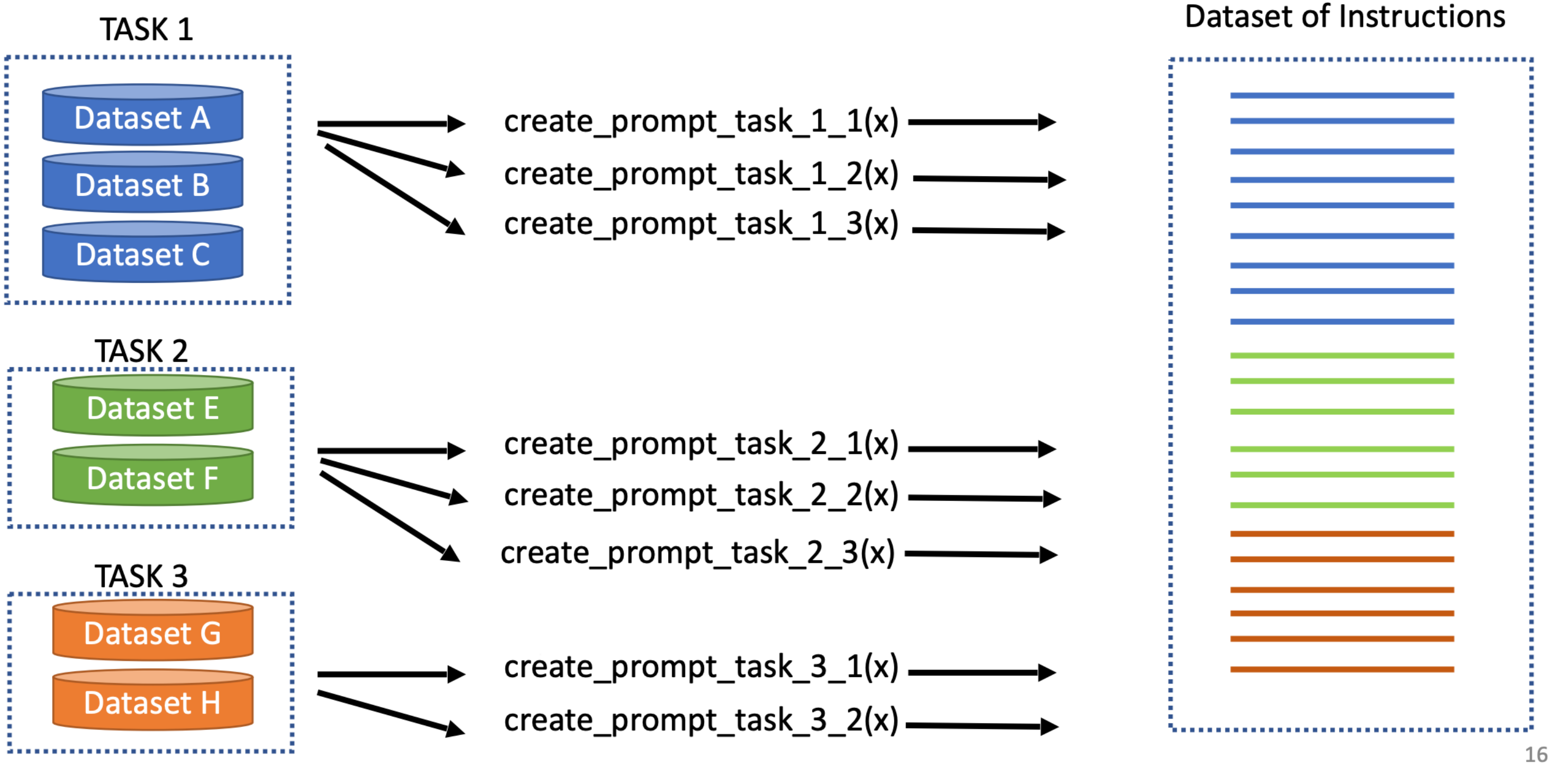
Diversity-inducing via Task Prompts

TASK 1 = Summarization



"Write highlights for this article:\n\n{text}\n\nHighlights: {highlights}"
"Write a summary for the following article:\n\n{text}\n\nSummary: {highlights}"
"{text}\n\nWrite highlights for this article. {highlights}"
"{text}\n\nWhat are highlight points for this article? {highlights}"
"{text}\nSummarize the highlights of this article. {highlights}"
"{text}\nWhat are the important parts of this article? {highlights}"
"{text}\nHere is a summary of the highlights for this article: {highlights}"
"Write an article using the following points:\n\n{highlights}\n\nArticle: {text}"
"Use the following highlights to write an article:\n\n{highlights}\n\nArticle:{text}"
"{highlights}\n\nWrite an article based on these highlights. {text}"

Diversity-inducing via Task Prompts



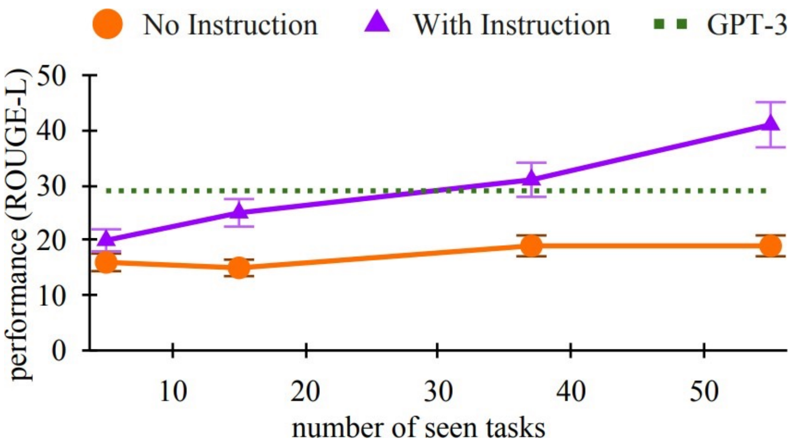
[Slide credit: Arman Cohan]

Release	Collection	Prompt Types	Tasks in Flan	# Exs	Methods
		Data Collection & Training Details			
2020 05	UnifiedQA	ZS	46 / 46	750k	
2021 04	CrossFit	FS	115 / 159	71.M	
2021 04	Natural Inst v1.0	ZS / FS	61 / 61	620k	+ Detailed k-shot Prompts
2021 09	Flan 2021	ZS / FS	62 / 62	4.4M	+ Template Variety
2021 10	P3	ZS	62 / 62	12M	+ Template Variety + Input Inversion
2021 10	MetaICL	FS	100 / 142	3.5M	+ Input Inversion + Noisy Channel Opt
2021 11	ExMix	ZS	72 / 107	500k	+ With Pretraining
2022 04	Super-Natural Inst.	ZS / FS	1556 / 1613	5M	+ Detailed k-shot Prompts + Multilingual
2022 10	GLM	FS	65 / 77	12M	+ With Pretraining + Bilingual (en, zh-cn)
2022 11	xP3	ZS	53 / 71	81M	+ Massively Multilingual
2022 12	Unnatural Inst.†	ZS	~20 / 117	64k	+ Synthetic Data
2022 12	Self-Instruct†	ZS	Unknown	82k	+ Synthetic Data + Knowledge Distillation
2022 12	OPT-IML Bench†	ZS + FS CoT	~2067 / 2207	18M	+ Template Variety + Input Inversion + Multilingual
2022 10	Flan 2022 (ours)	ZS + FS CoT	1836	15M	+ Template Variety + Input Inversion + Multilingual

The Flan Collection: Designing Data and Methods for Effective Instruction Tuning (Longpre et al., 2023)

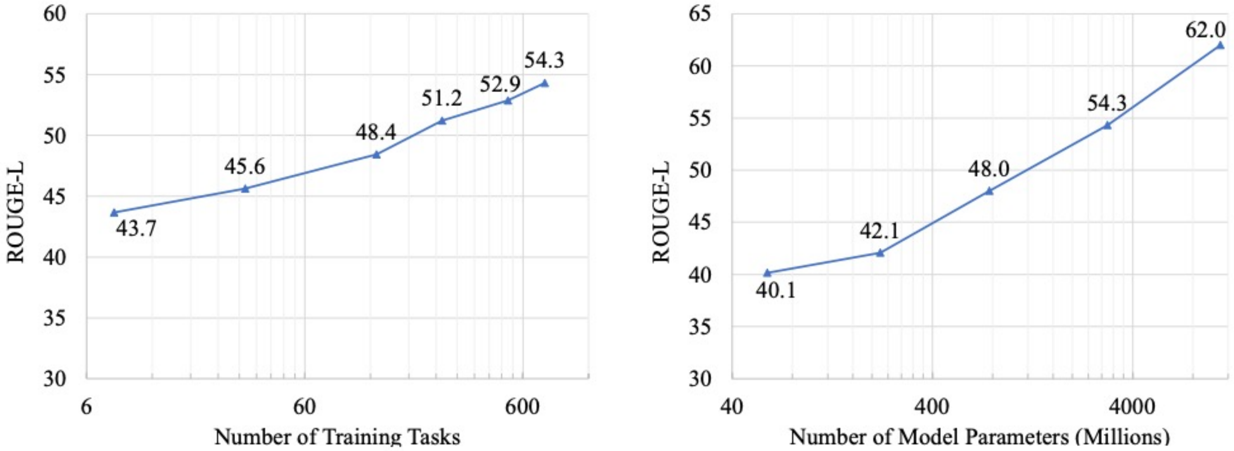
Scaling Instruction-Tuning

Linear growth of model performance with exponential increase in observed tasks.



Cross-Task Generalization via Natural Language Crowdsourcing Instructions (Mishra et al., 2022)

Scaling Instruction-Tuning



Linear growth of model performance
with exponential increase in observed tasks and model size.

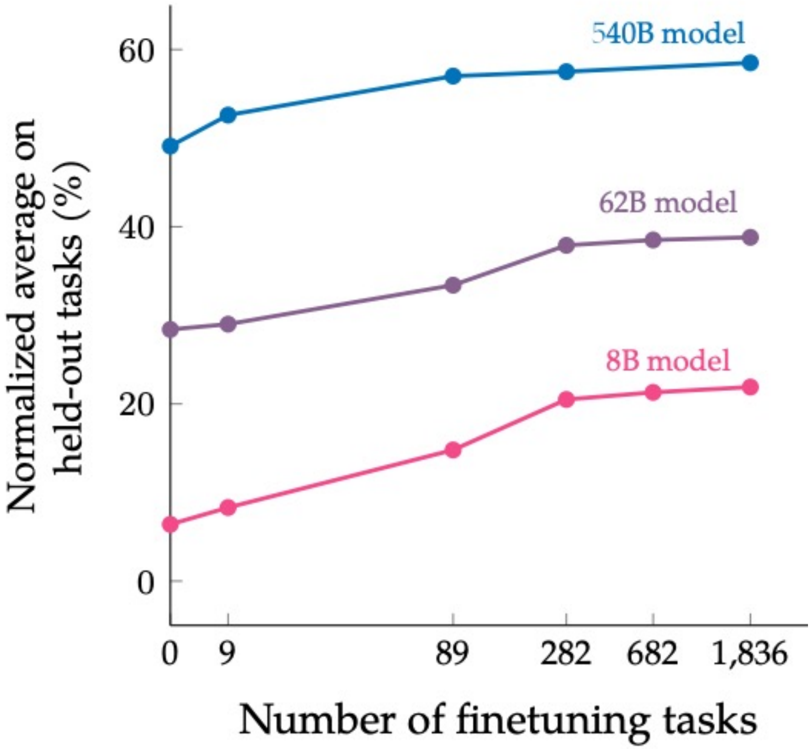
[Super-NaturalInstructions: Generalization via Declarative Instructions on 1600+ NLP Tasks, Wang et al. 2022]

Scaling Instruction-Tuning

- Instruction finetuning improves performance by a large margin compared to no finetuning

- Increasing the number of finetuning tasks improves performance

- Increasing model scale by an order of magnitude (i.e., 8B ⟶ 62B or 62B ⟶ 540B) improves performance substantially for both finetuned and non-finetuned models



[Scaling Instruction-Finetuned Language Models, Chung et al. 2022]

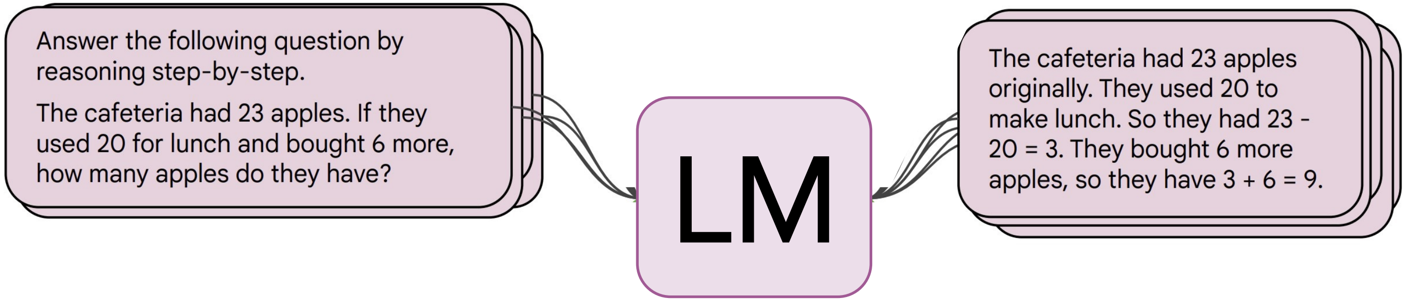
Instruction tuning doesn't have significant cost compared with pretraining

Params	Model	Architecture	Pre-training Objective	Pre-train FLOPs	Finetune FLOPs	% Finetune Compute
80M	Flan-T5-Small	encoder-decoder	span corruption	1.8E+20	2.9E+18	1.6%
250M	Flan-T5-Base	encoder-decoder	span corruption	6.6E+20	9.1E+18	1.4%
780M	Flan-T5-Large	encoder-decoder	span corruption	2.3E+21	2.4E+19	1.1%
3B	Flan-T5-XL	encoder-decoder	span corruption	9.0E+21	5.6E+19	0.6%
11B	Flan-T5-XXL	encoder-decoder	span corruption	3.3E+22	7.6E+19	0.2%
8B	Flan-PaLM	decoder-only	causal LM	3.7E+22	1.6E+20	0.4%
62B	Flan-PaLM	decoder-only	causal LM	2.9E+23	1.2E+21	0.4%
540B	Flan-PaLM	decoder-only	causal LM	2.5E+24	5.6E+21	0.2%
62B	Flan-cont-PaLM	decoder-only	causal LM	4.8E+23	1.8E+21	0.4%
540B	Flan-U-PaLM	decoder-only	prefix LM + span corruption	2.5E+23	5.6E+21	0.2%

Scaling Instruction-Finetuned Language Models (2022)

Limits of Instruction-Tuning

1. Difficult to collect diverse data.
2. Resulting models may not be good at open-ended generation tasks.
   - Incentivizes word-by-word rote learning => The resulting LM's generality/creativity is bounded by that of their supervision data.



Limits of Instruction-Tuning

1. Difficult to collect diverse data.
2. Resulting models may not be good at open-ended generation tasks.
   - Incentivizes word-by-word rote learning => The resulting LM's generality/creativity is bounded by that of their supervision data.
3. Resulting models may hallucinate more regularly.
   - Labeled data is collected agnostic to the LM's knowledge => there might be a mismatch between labeled data and LM knowledge.
   - Hence, we may be encouraging "hypocritic" behavior => further hallucinations

Summary Thus Far

- **Instruction-tuning:** Training LMs with annotated input instructions and their output.
  - Improves performance of LM's zero-shot ability in following instructions.
  - Scaling the instruction tuning data size improves performance.
  - Diversity of prompts is crucial.
  - Compared with pretraining, instruction tuning has a minor cost (Typically consumes <1% of the total training budget)
- Cons:
  - It's expensive to collect ground- truth data for tasks.
  - This is particularly difficult for open-ended creative generation have no right answer.
  - Prone to hallucinations.

[Weller et al. 2020. Mishra et al. 2021; Wang et al. 2022, Sanh et al. 2022; Wei et al., 2022, Chung et al. 2022, many others]

Aligning Language Models: Reinforcement Learning w/ Feedback

Why Reinforcement Learning?

- Remember the limits of Instruction-tuning?

  1. Difficult to collect diverse labeled data
  2. Rote learning (token by token) —
     - limited creativity
  3. Agnostic to model's knowledge —
     - may encourage hallucinations

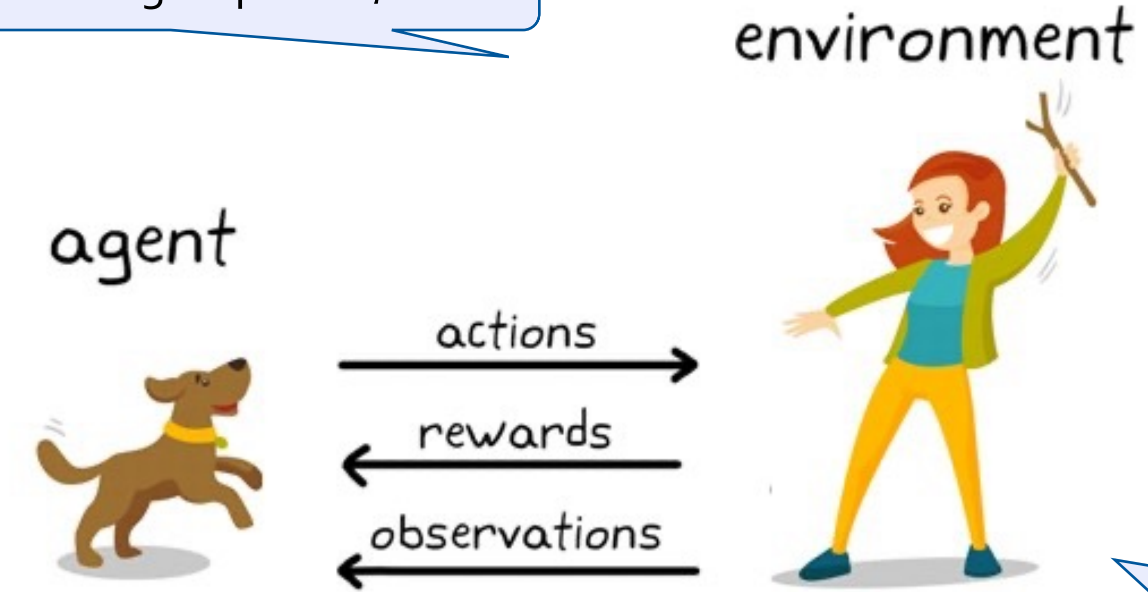
Limited/sparse feedback—usually considered a curse, but now a blessing.

"don't give a man fish rather teach him how to fish by himself"

The model itself should be involved in the alignment loop.

Reinforcement Learning: Intuition

Action here: generating responses/token

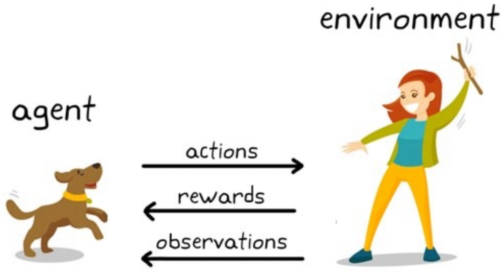


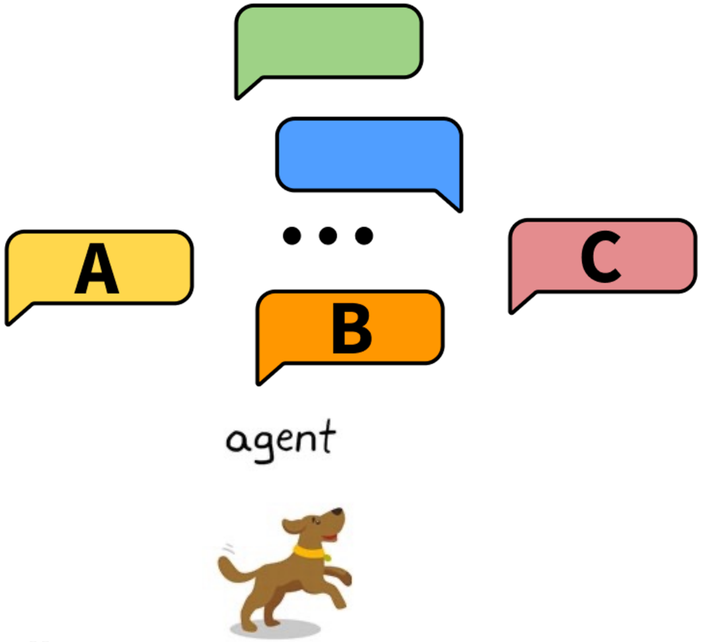
Reward here: whether humans liked the generation (sequence of actions=tokens)

[figure credit]

Intuition

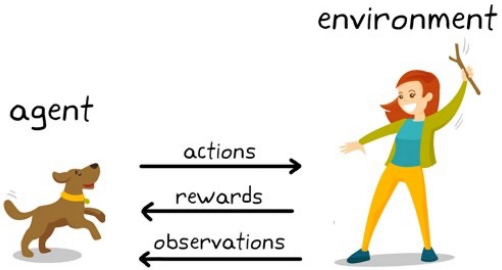
Task: choose the better next message in a conversation

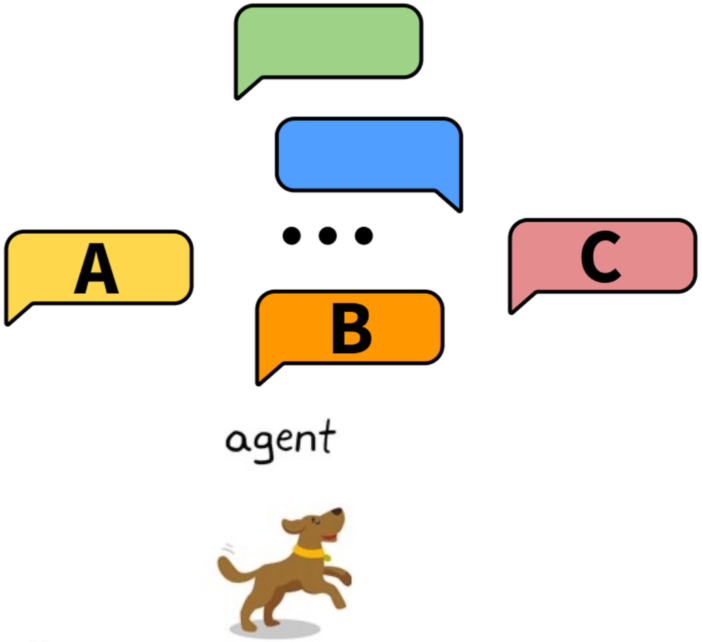


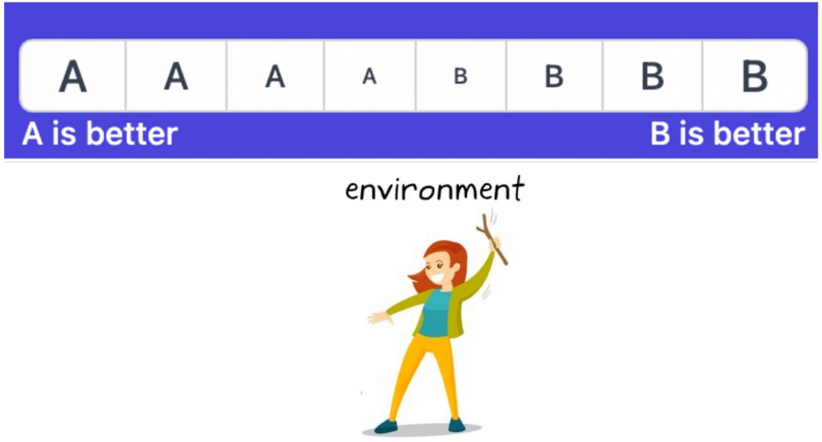


Intuition

Scoring interface: Likert scale or rankings







Intuition

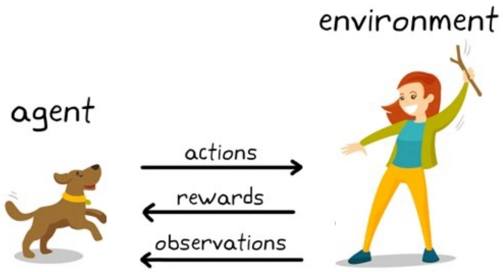
Human
I have to come up with a conversation with an AI assistant for the purposes of showing it off in a demo. What sorts of conversation topics would you recommend?

Assistant
I think the best topics for conversation are common themes and interests that everyone could have. Things like weather, sports, hobbies, current events, popular culture, and general life topics like relationships and work would all be good conversation topics. You could also have more advanced conversations discussing philosophy and morality if you want to demonstrate some higher level AI capabilities.

Human
I think I'm more interested in an advanced conversation. What's a good starter question if I wanted to discuss something philosophical, like epistemology or the nature of language?

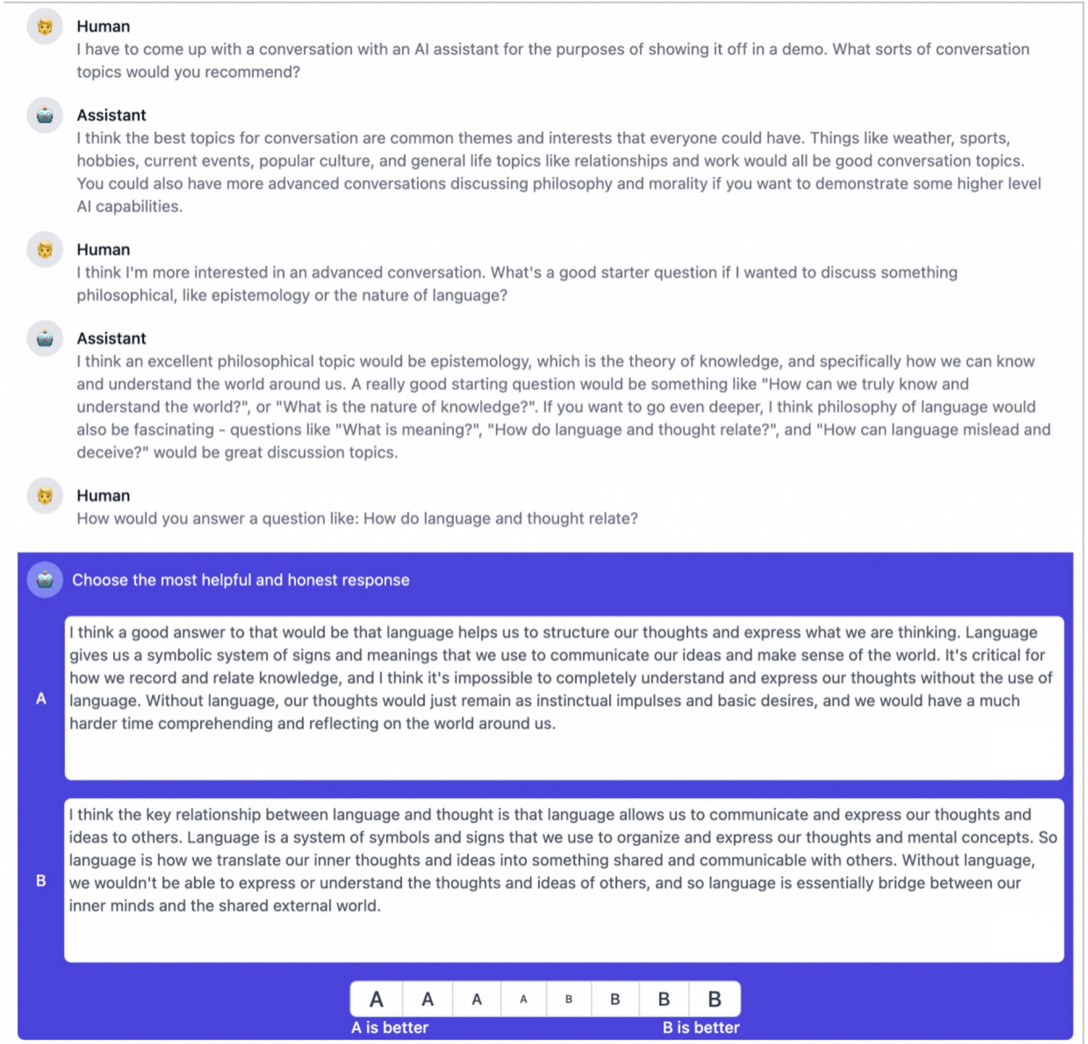
Assistant
I think an excellent philosophical topic would be epistemology, which is the theory of knowledge, and specifically how we can know and understand the world around us. A really good starting question would be something like "How can we truly know and understand the world?", or "What is the nature of knowledge?". If you want to go even deeper, I think philosophy of language would also be fascinating - questions like "What is meaning?", "How do language and thought relate?", and "How can language mislead and deceive?" would be great discussion topics.

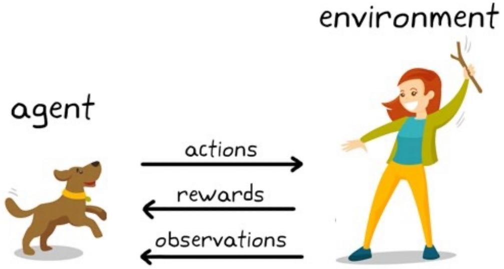
human has conversation with the LLM



Intuition

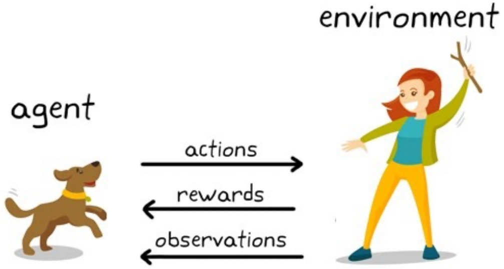
LLM provides two options for next responses

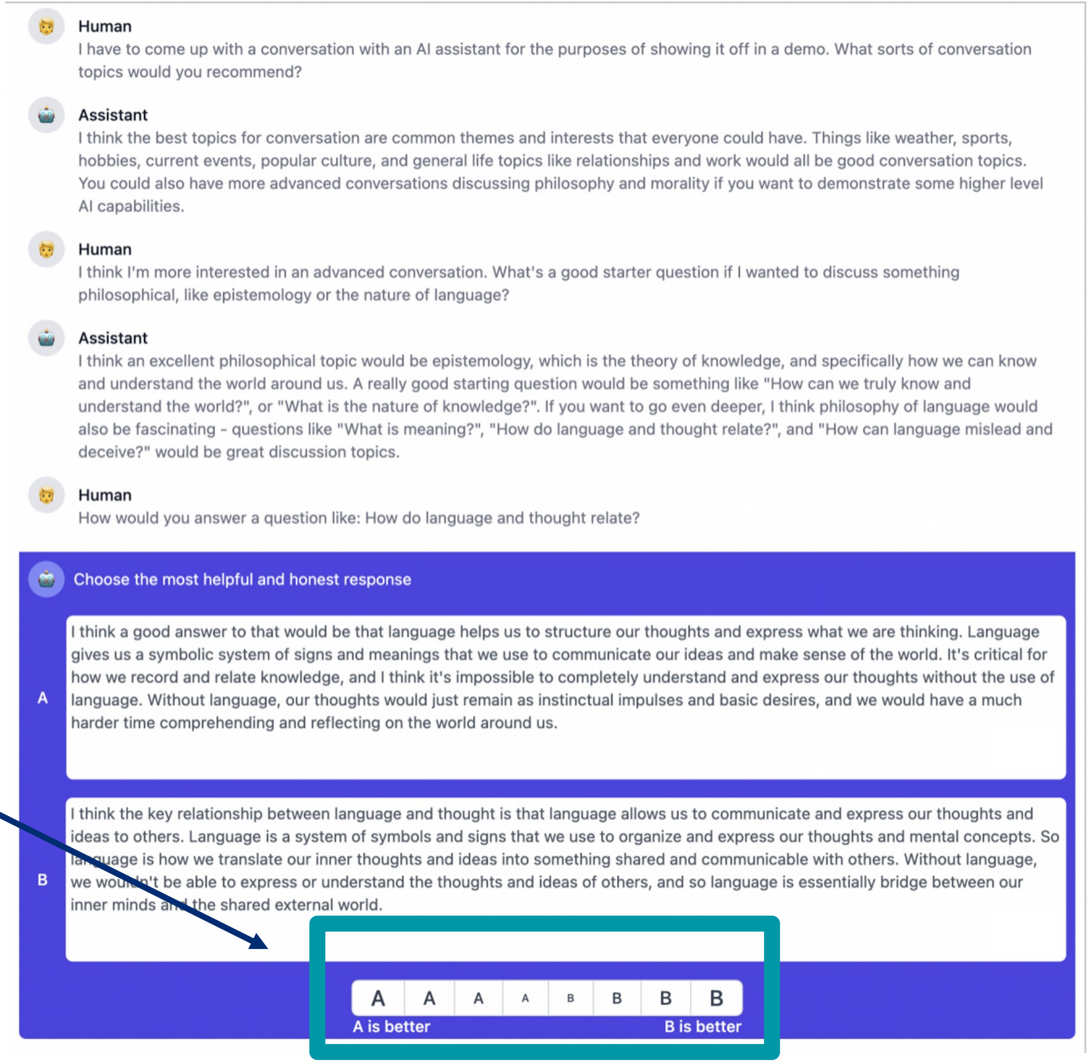




Intuition

human rates better response





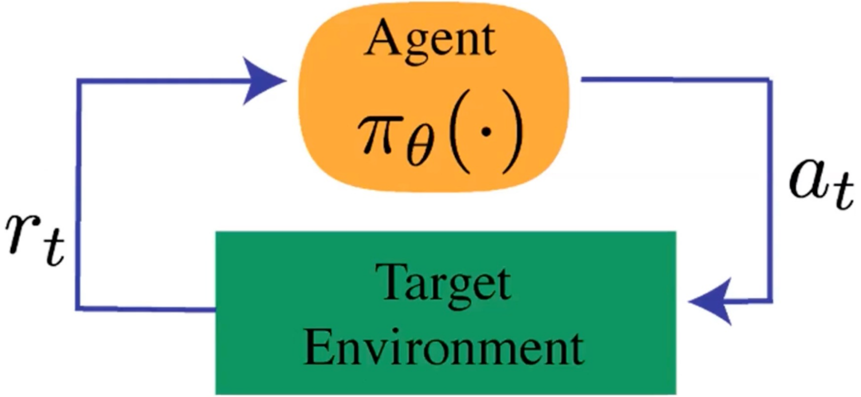
Reinforcement Learning: Abridged History

- The field of reinforcement learning (RL) has studied these (and related) problems for many years now [Williams, 1992; Sutton and Barto, 1998]
- Circa 2013: resurgence of interest in RL applied to deep learning, game-playing [Mnih et al., 2013]
- But there is a renewed interest in applying RL. Why?
  - RL w/ LMs has commonly been viewed as very hard to get right (still is!)
  - We have found successful RL variants that work for language models (e.g., PPO; [Schulman et al., 2017])

AlphaGo

Reinforcement Learning: Formalism

- An agent interacts with an environment by taking actions
- The environment returns a reward for the action and a new state (representation of the world at that moment).
- Agent uses a policy function to choose an action at a given state.
- We need to figure out: (1) reward function and (2) the policy function

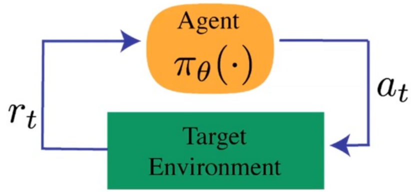


Some notation:

r_t : reward
a_t : action
$a_t \sim \pi_\theta(s_t)$: policy

[Fig credit: Nate Lambert]

Reinforcement Learning from Human Feedback



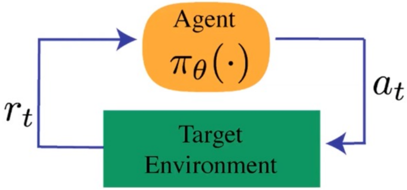
- Imagine a reward function: $R(s; \text{promp}) \in \mathbb{R}$ for any output s to a prompt.
- The reward is higher when humans prefer the output.
- Good generation is equivalent to finding reward-maximizing outputs:

Expected reward over the course of sampling from our policy (generative model)

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

$p_\theta(s)$ is a pre-trained model with params θ we would like to optimize (policy function)

Reinforcement Learning from Human Feedback



- Imagine a reward function: $R(s;\text{prompt}) \in \mathbb{R}$ for any output s to a prompt.
- The reward is higher when humans prefer the output.
- Good generation is equivalent to finding reward-maximizing outputs:

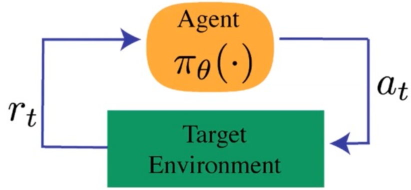
Expected reward over the course of sampling from our policy (generative model)

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

$p_\theta(s)$ is a pre-trained model with params θ we would like to optimize (policy function)

- On the notation:
  - "$\mathbb{E}$" here is an empirical expectation (i.e., average).
  - "~" indicates sampling from a given distribution.

Reinforcement Learning from Human Feedback



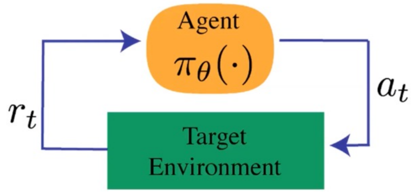
- Imagine a reward function: $R(s; \text{prompt}) \in \mathbb{R}$ for any output s to a prompt.
- The reward is higher when humans prefer the output
- Good generation is equivalent to finding reward-maximizing outputs:

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

- What we need to do:
  - (1) Estimate the reward function $R(s; \text{prompt})$.
  - (2) Find the best generative model p_θ that maximizes the expected reward:

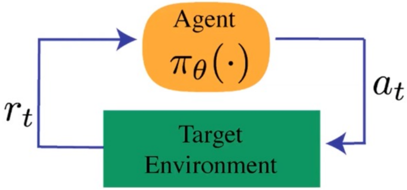
$$\hat{\theta} = \text{argmax}_\theta \mathbb{E}_{\hat{s} \sim p_\theta}[R(\hat{s}; \text{prompt})]$$

Step 1: Estimating the Reward R



- Obviously, we don't want to use human feedback directly since that could be 💰💰💰
- Alternatively, we can build a model to mimic their preferences [Knox and Stone, 2009]

Step 1: Estimating the Reward R



- Obviously, we don't want to use human feedback directly since that could be 💰💰💰
- Alternatively, we can build a model to mimic their preferences [Knox and Stone, 2009]
- Approach 1: get humans to provide absolute scores for each output

Challenge: human judgments on different instances and by different people can be noisy and mis-calibrated!

prompt

Explain "space elevators" to a 6-year-old.

LM

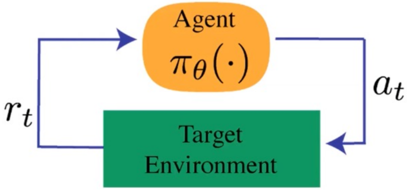
p_θ

s_1 It is like any typical elevator, but it goes to space. … 👩 → 0.8

s_2 Explain gravity to a 6-year-old. … 👨🏿 → 1.2

$s_1, s_2 \sim p_\theta$

Step 1: Estimating the Reward R



- Obviously, we don't want to use human feedback directly since that could be 💰💰💰
- Alternatively, we can build a model to mimic their preferences [Knox and Stone, 2009]
- Approach 2: ask for pairwise comparisons [Phelps et al. 2015; Clark et al. 2018]

Bradley-Terry [1952]
paired comparison model

Pairwise comparison of multiple provides which can be more reliable

prompt

Explain "space elevators" to a 6-year-old.

LM

p_θ

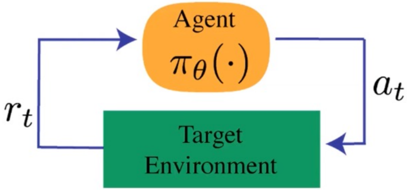
s_1 It is like any typical elevator, but it goes to space. … 👍

s_2 Explain gravity to a 6-year-old. … 👎

👩

$s_1, s_2 \sim p_\theta$

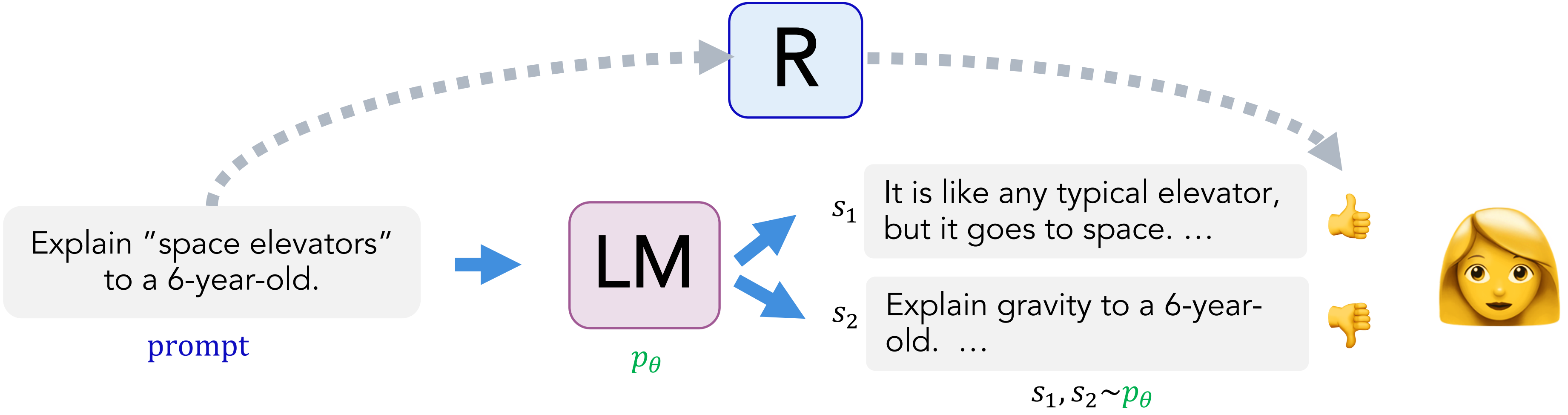
Step 1: Estimating the Reward R



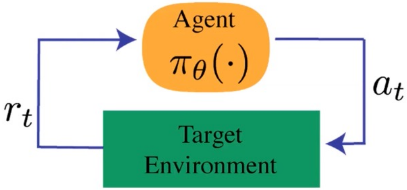
$$J(\phi) = -\mathbb{E}_{(s^+, s^-)}\left[\log \sigma\left(R(s^+; \text{prompt}) - R(s^-; \text{prompt})\right)\right]$$

"winning" sample

"losing" sample



Step 1: Estimating the Reward R



$$J(\phi) = -\mathbb{E}_{(s^+, s^-)}\left[\log \sigma\left(R(s^+; \text{prompt}) - R(s^-; \text{prompt})\right)\right]$$

"winning" sample

"losing" sample

The reward model returns a scalar reward which should numerically represent the human preference.

R

Explain "space elevators" to a 6-year-old.

prompt

LM

p_θ

s_1 It is like any typical elevator, but it goes to space. …

s_2 Explain gravity to a 6-year-old. …

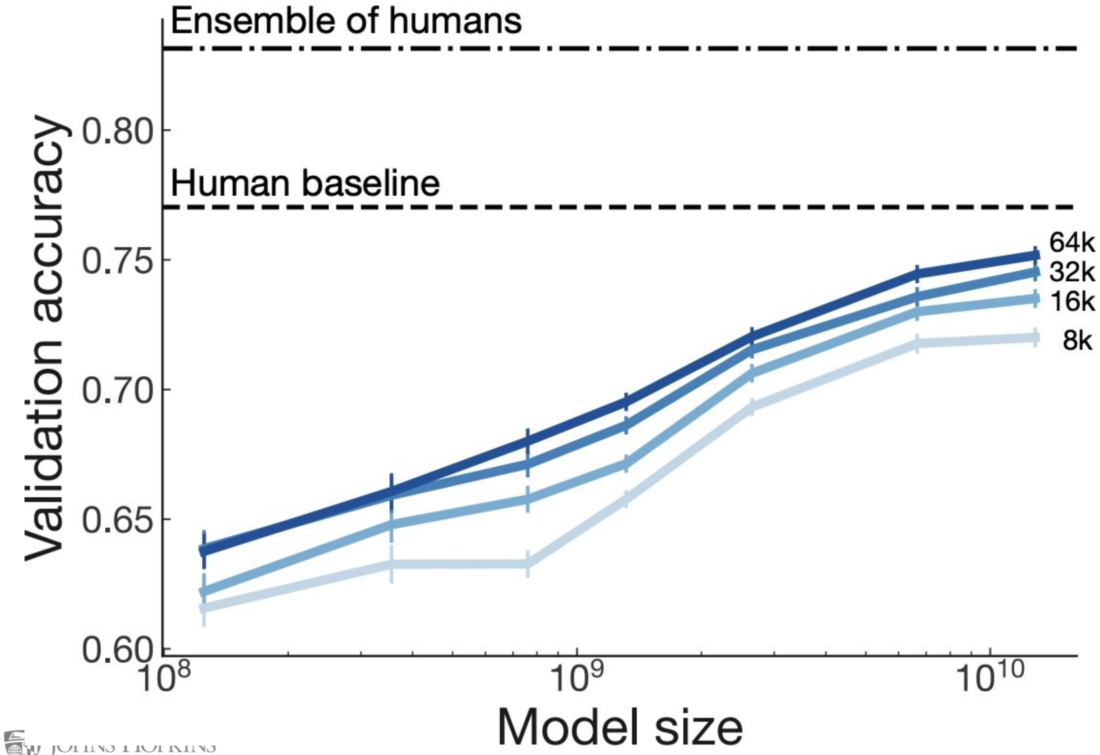
$s_1, s_2 \sim p_\theta$

$R(s_2; \text{prompt}) = 1.2$

$R(s_1; \text{prompt}) = 0.8$

Scaling Reward Models

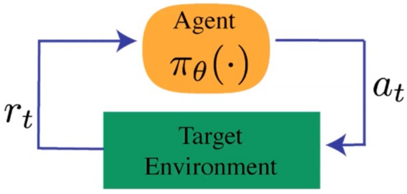
R



Large enough reward trained on large enough data approaching human performance.

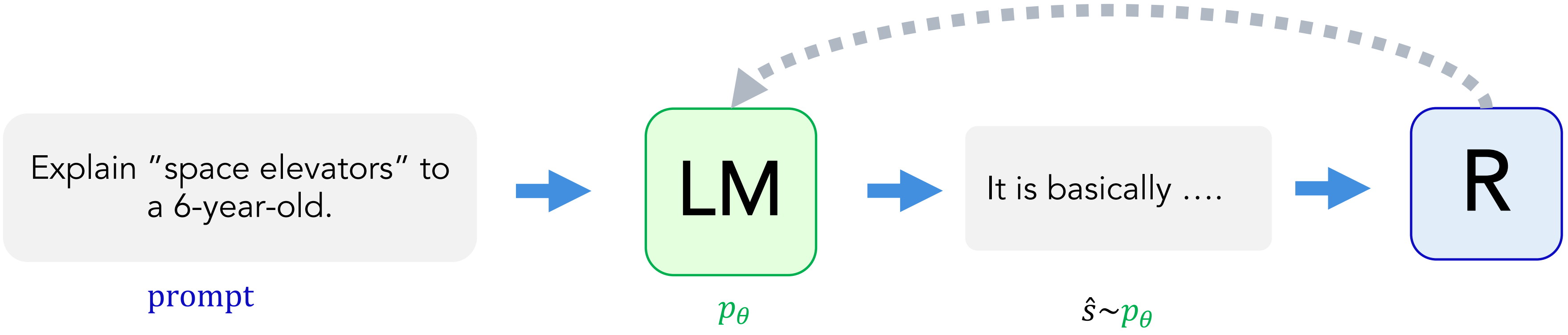
[Stiennon et al., 2020]

Step 2: Optimizing the Policy Function

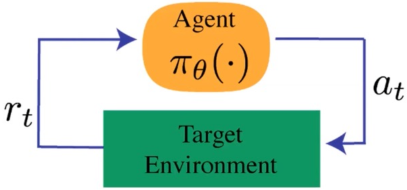


- Policy function := The model that makes decisions (here, generates responses)
- How do we change our LM parameters θ to maximize this?

$$\hat{\theta} = \text{argmax}_{\theta}\ \mathbb{E}_{\hat{s} \sim p_\theta}[R(\hat{s}; \text{prompt})]$$



Step 2: Optimizing the Policy Function



- Policy function := The model that makes decisions (here, generates responses)
- How do we change our LM parameters θ to maximize this?

$$\hat{\theta} = \mathrm{argmax}_\theta \; \mathbb{E}_{\hat{s}\sim p_\theta}[R(\hat{s};\text{prompt})]$$

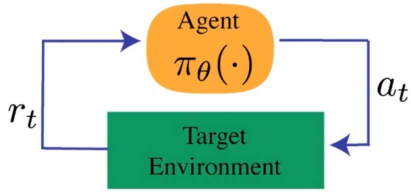
- Let's try doing gradient ascent!

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

How do we estimate this expectation?

- Turns out that we can write this "gradient of expectation" to a simpler form.

Policy Gradient [Williams, 1992]



- How do we change our LM parameters θ to maximize this?

$$\hat{\theta} = \operatorname{argmax}_\theta \, \mathbb{E}_{\hat{s} \sim p_\theta}[R(\hat{s}; \text{prompt})]$$

- Let's try doing gradient ascent!

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

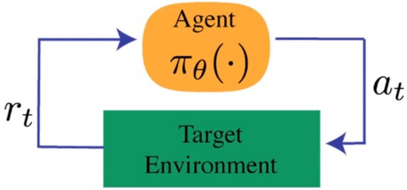
- With a bit of math, this can be approximated as Monte Carlo samples from $p_\theta(s)$:

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

Proof next slide; check it later in your own time!

- This is "policy gradient", an approach for estimating and optimizing this objective.
- Oversimplified. For full treatment of RL see 701.741 course, or Huggingface's course

Derivations (check it later in your own time!)



- Let's compute the gradient:

Def. of "expectation"

Gradient distributes over sum

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

- Log-derivative trick $\nabla_\theta p_\theta(s) = p_\theta(s)\,.\,\nabla_\theta \log p_\theta(s)$ to turn sum back to expectation:

Log-derivative trick

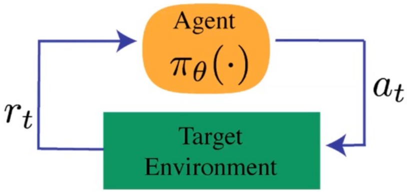
$$\nabla_\theta \mathbb{E}_{s\sim p_\theta(s)}[R(s;p)] = \sum_s R(s;p)\, p_\theta(s)\, \nabla_\theta \log p_\theta(s) = \mathbb{E}_{s\sim p_\theta(s)}[R(s;p)\, \nabla_\theta \log p_\theta(s)]$$

- Approximate this expectation with Monte Carlo samples from $p_\theta(s)$:

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

JOHNS HOPKINS WHITING SCHOOL of ENGINEERING

Policy Gradient [Williams, 1992]



Note, $R(s;p)$ could be any arbitrary, non-differentiable reward function that we design.

- This gives us the following update rule:

$$\theta_{t+1} \leftarrow \theta_t + \alpha \frac{1}{n} \sum_{i=1}^{n} R(s;p)\, \nabla_\theta \log p_\theta(s)$$

- If $R(s;p)$ is large, we take proportionately large steps to maximize $p_\theta(s)$
- If $R(s;p)$ is small, we take proportionately small steps to maximize $p_\theta(s)$

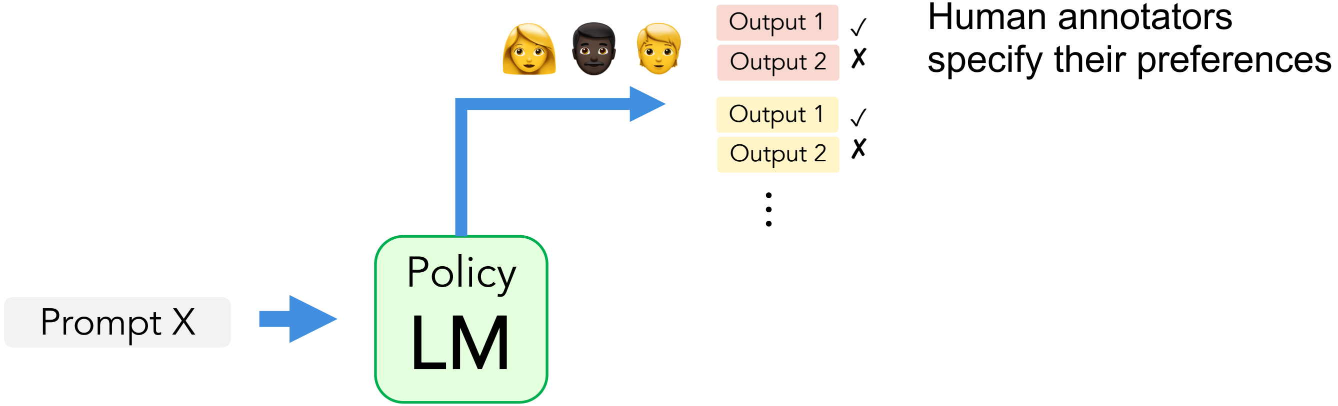
This is why it's called "reinforcement learning":
we reinforce good actions, increasing the chance they happen again.

Johns Hopkins Whiting School of Engineering

[Slide credit: Jesse Mu]

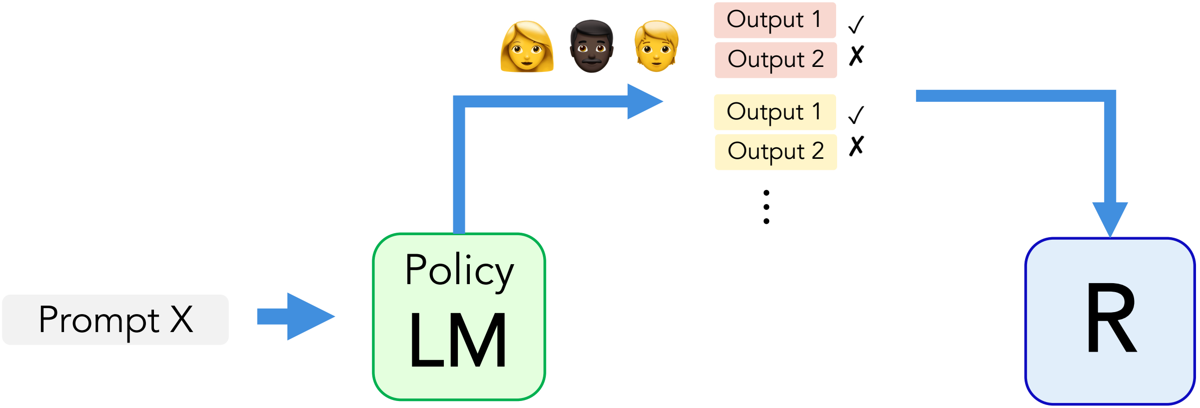
Putting it Together

- First collect a dataset of human preferences
  - Present multiple outputs to human annotators and ask them to rank the output based on preferability



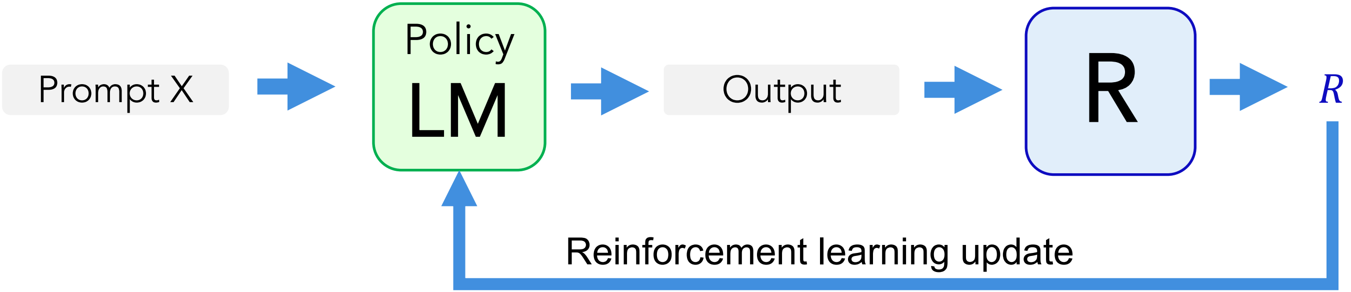
Putting it Together (2)

- Using this data, we can train a reward model
  - The reward model returns a scalar reward which should numerically represent the human preference.



Putting it Together (3)

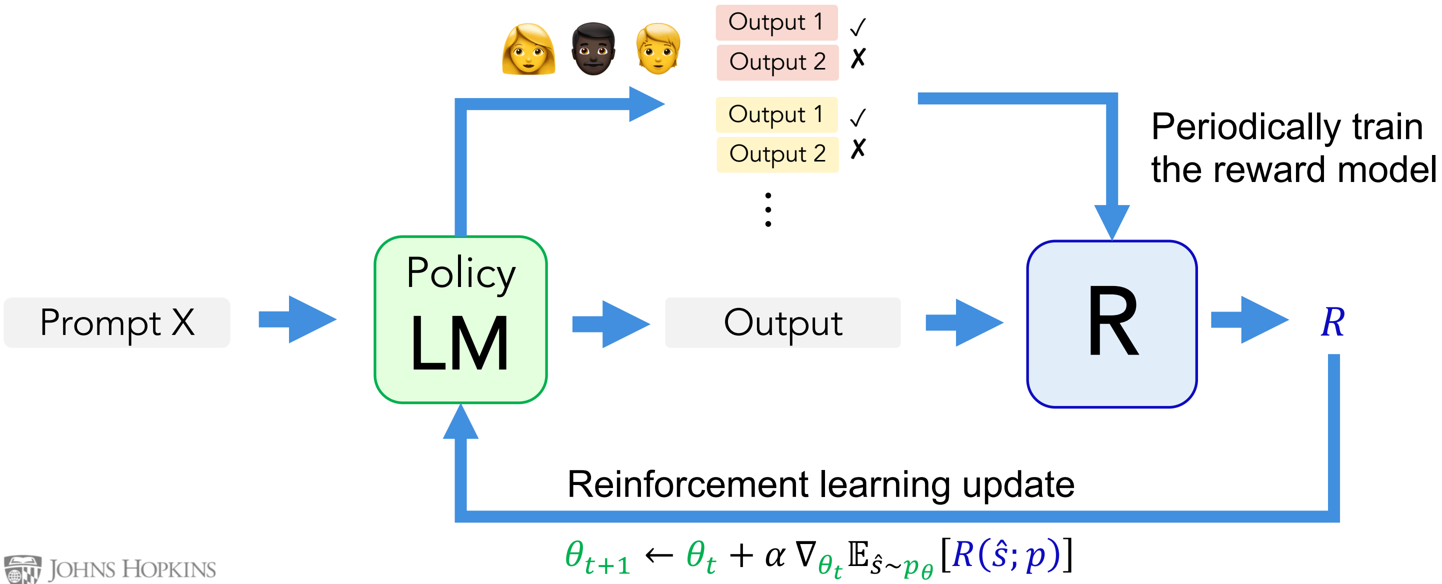
- We want to learn a policy (a Language Model) that optimizes against the reward model



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

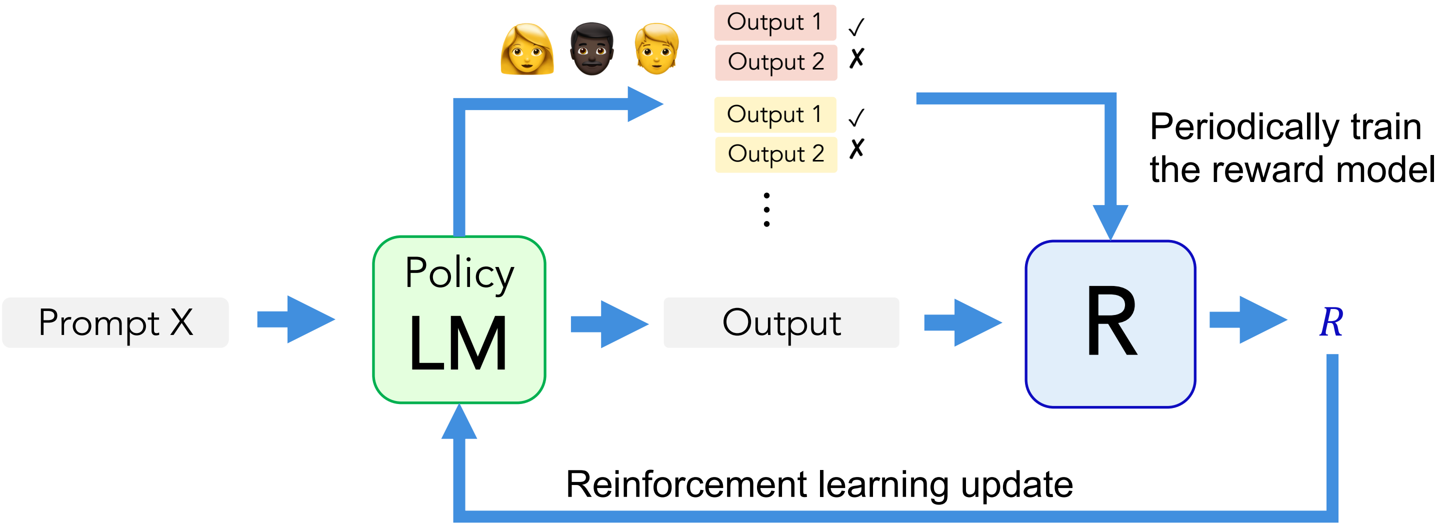
Putting it Together (4)

- Periodically train the reward model with more samples and human feedback



One missing ingredient

- It turns out that this approach doesn't quite work. (Any guesses why?)
  - The policy will learn to "cheat".

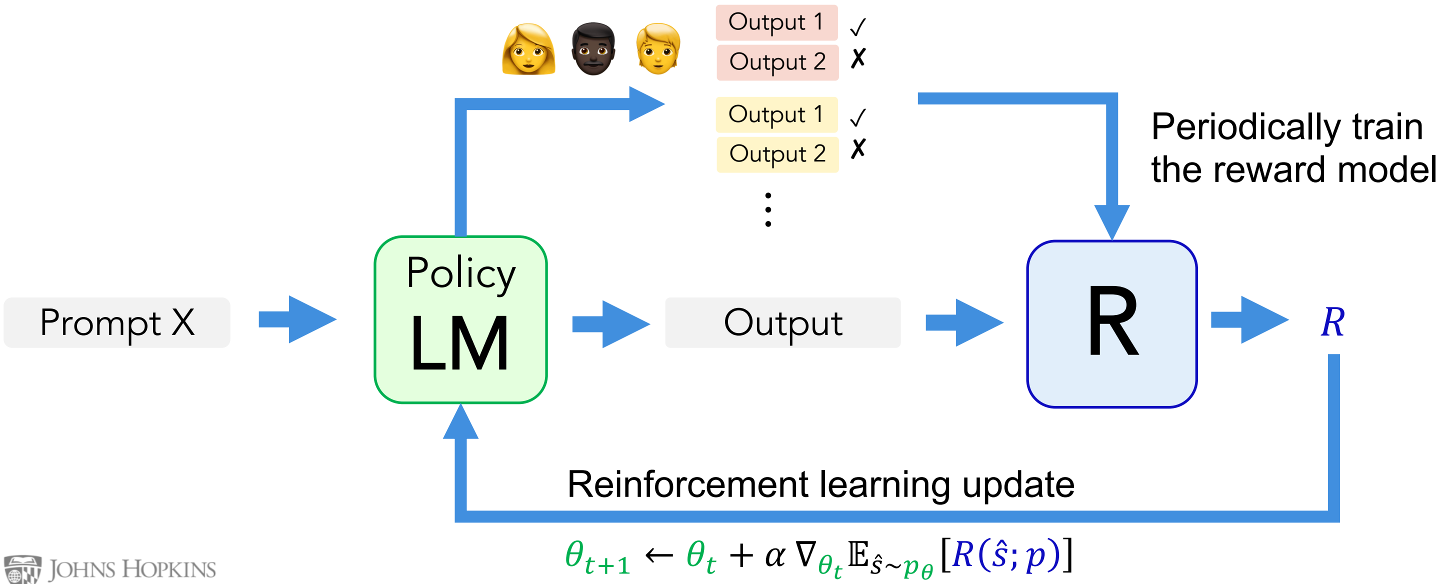


$$\theta_{t+1} \leftarrow \theta_t + \alpha \nabla_{\theta_t} \mathbb{E}_{\hat{s} \sim p_\theta}[R(\hat{s}; p)]$$

One missing ingredient

How do you resolve this? 🤔

- Will learn to produce an output that would get a **high** reward but is **gibberish** or **irrelevant** to the prompt.
- Note, since $R(s; p)$ is trained on natural inputs, it may not generalize to unnatural inputs.



Regularizing with Pre-trained Model

- **Solution:** add a penalty term that penalizes too much deviations from the distribution of the pre-trained LM.

$$\hat{R}(s;p) := R(s;p) - \beta \log\left(\frac{p^{RL}(s)}{p^{PT}(s)}\right)$$

pay a price when $p^{RL}(s) > p^{PT}(s)$

- This prevents the policy model from diverging too far from the pretrained model.
- The above regularization is equivalent to adding a KL-divergence regularization term. You will see/prove the details in HW7!!

RLHF: Putting it All Together [Stiennon et al. 2020]

1. Select a pre-trained generative model as your base: $p_\theta^{PT}(s)$
2. Build a reward model $R(s; p)$ that produces scalar rewards for outputs, trained on a dataset of human comparisons
3. Regularize the reward function: $$\hat{R}(s; p) \coloneqq R(s; p) - \beta \log\left(\frac{p^{RL}(s)}{p^{PT}(s)}\right)$$
4. Iterate:
    1. Fine-tune the policy $p_\theta^{RL}(s)$ to maximize our reward model $R(s; p)$
    $$\theta_{t+1} \leftarrow \theta_t + \alpha \frac{1}{n} \sum_{i=1}^{n} \hat{R}(s; p)\, \nabla_\theta \log p_\theta^{RL}(s)$$
    2. Occasionally repeat repeat 2-3 to update the reward model.

The overall recipe 👨‍🍳

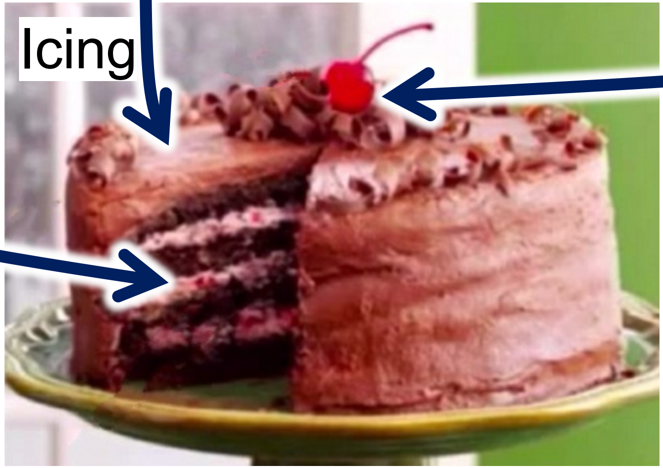
Pre-train → Align (instruct-tune) → Align (RLHF)

The overall recipe 🧑‍🍳

Pre-train → Align (instruct-tune) → Align (RLHF)

The overall recipe 👨‍🍳: Yann’s Three-layered cake

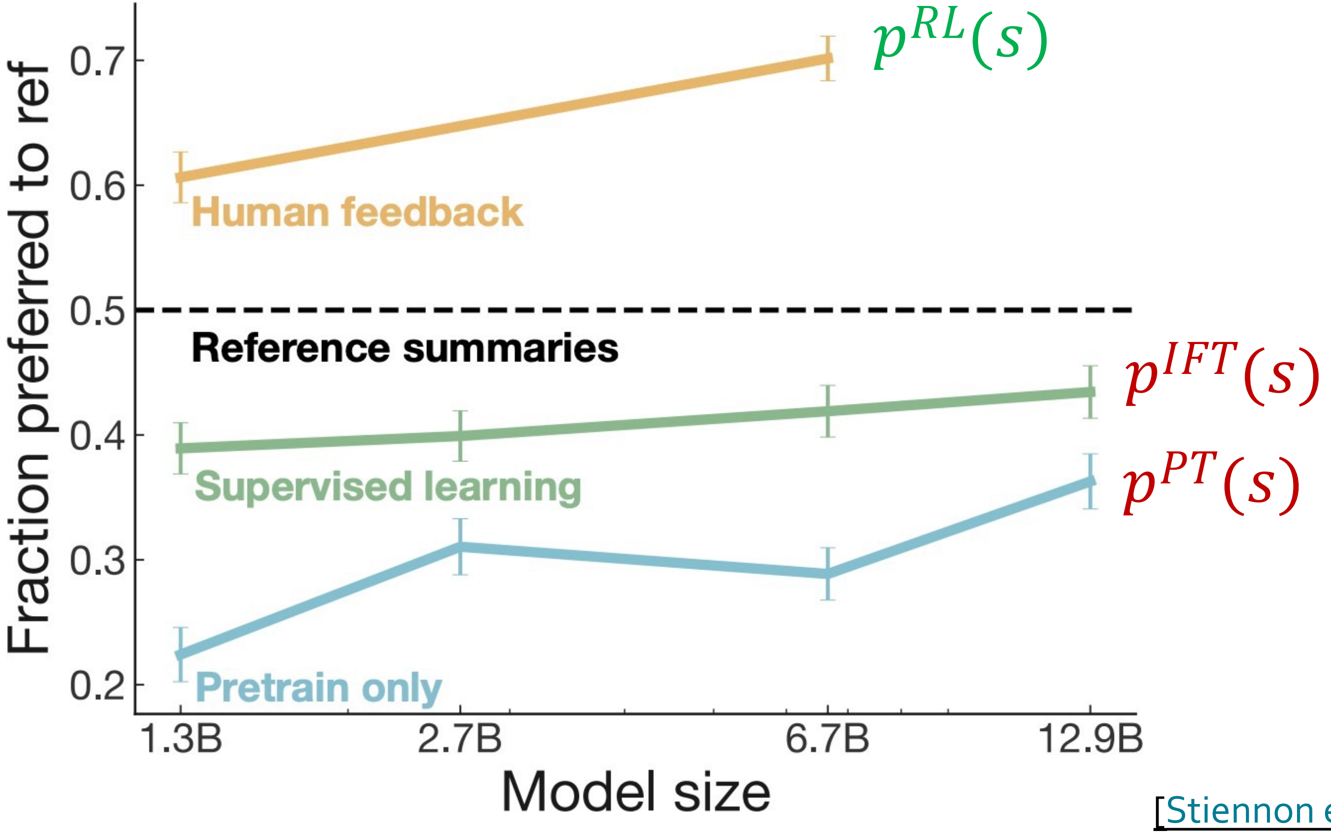
Pre-train → Align (instruct-tune) → Align (RLHF)



Cake génoise

Cherry on the cake

RLHF Gains over Instruction-Tuning



[Stiennon et al., 2020]

GPT3 vs. InstructGPT3 (RLHF-ed)

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.
```

InstructGPT

```
People went to the moon, and they took pictures of what they saw, and
sent them back to the earth so we could all see them.
```

GPT3 vs. InstructGPT3 (RLHF-ed)

PROMPT *Write a short poem about a wise frog.*

COMPLETION

GPT-3

```
Write a short story in which a character has two different names.

Write a short story in which you try to get something back that you
have lost.

Write a short story in which a character has a bad dream.
```

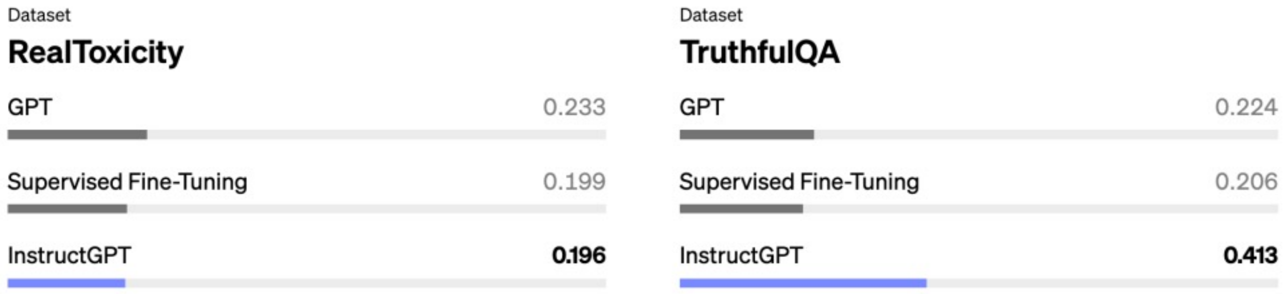
InstructGPT

```
The frog is a symbol of wisdom
He knows all the secrets of the world
He is a master of disguise
And a great teacher of life
He is a symbol of transformation
And the bringer of change
He is the frog who has seen it all
And knows the meaning of it all
```

Can Help with Toxicity and Truthfulness

- Note, reward model can be used to induce any desired behavior as needed:
  - Avoiding bias
  - Avoiding responses outside its scope
  - Avoiding toxicity
  - …

Lower is better



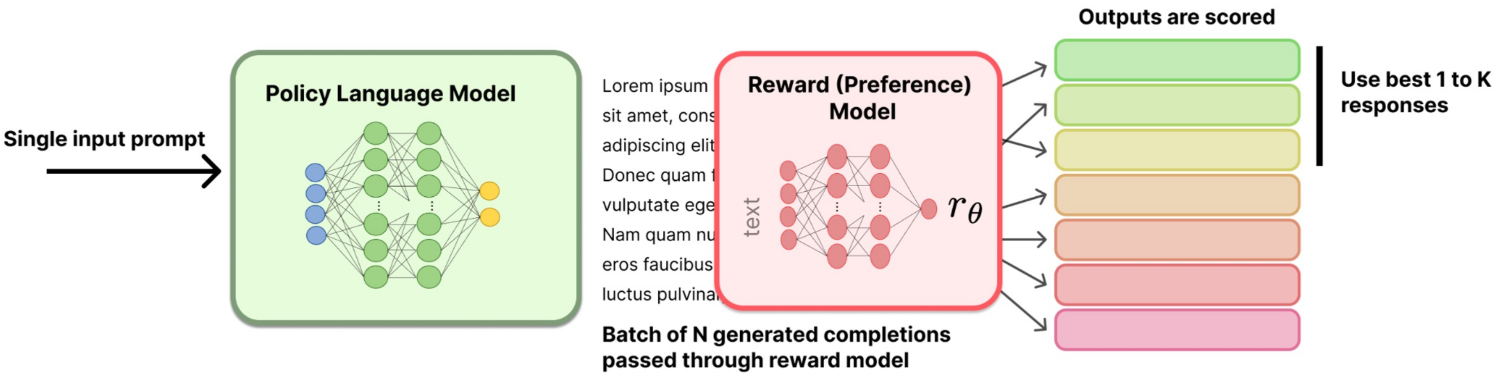
Higher is better

GPT-4: Demystifying the Details

- It's opaque, but now we can make educated guesses.
  - "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 [human] 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."
  - "To create a reward model for reinforcement learning, we needed to collect comparison data, which consisted of two or more model responses ranked by quality. To collect this data, we took conversations that AI trainers had with the chatbot. We randomly selected a model-written message, sampled several alternative completions, and had AI trainers rank them."
  - "Using these reward models, we can fine-tune the model using Proximal Policy Optimization. We performed several iterations of this process."

Best-of-N Sampling Algorithm

- Best-of-N:
  - Sample N outputs from policy
  - Score them all with the reward
- Example usage: https://huggingface.co/docs/trl/main/en/best_of_n



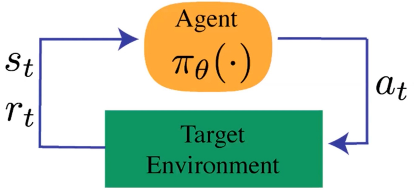
Summary Thus Far

- Reinforcement learning can help mitigate some of the problems with supervised instruction tuning
- RLHF uses two models
  - Reward model is trained via ranking feedback of humans.
  - Policy model learns to generate responses that maximize the reward model.

- Limitations:
  - RL can be tricky to get right
  - Training a good reward may require a lot of annotations

There are plenty of RL variants out there …

Reinforcement Learning: Families



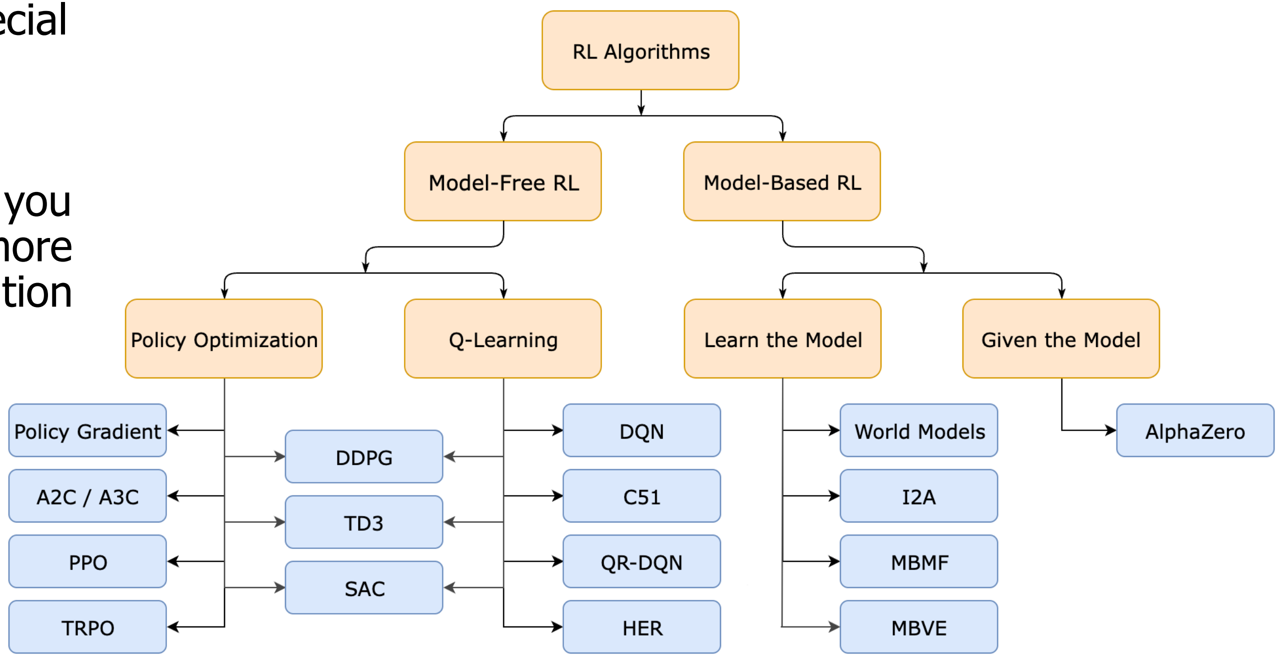
There are a variety of RL algorithms (out of scope for us). Broadly,

- **Policy-Based Methods**, learn a policy function directly.
  - Takes a state as input and outputs an action (or a distribution over
    - actions) to take.
  - We're not too concerned with determining the value or "goodness" of each state-action pair
  - We just want to know what to do in each state to perform well.
- **Value-based methods**:
  - the idea is to find the value of each state or state-action pair, and then act in a way that maximizes these values.

The variant we just saw

The Bigger Picture

- What we saw was a special case of PPO algorithm.
- In HW7, we have given you the code for a slightly more complex policy optimization (PPO)!



https://spinningup.openai.com/en/latest/spinningup/rl_intro2.html

Aligning Language Models: Failures and Challenges

RL Failure Modes

- Can be quite tricky to get right …

The 37 Implementation Details of Proximal Policy Optimization

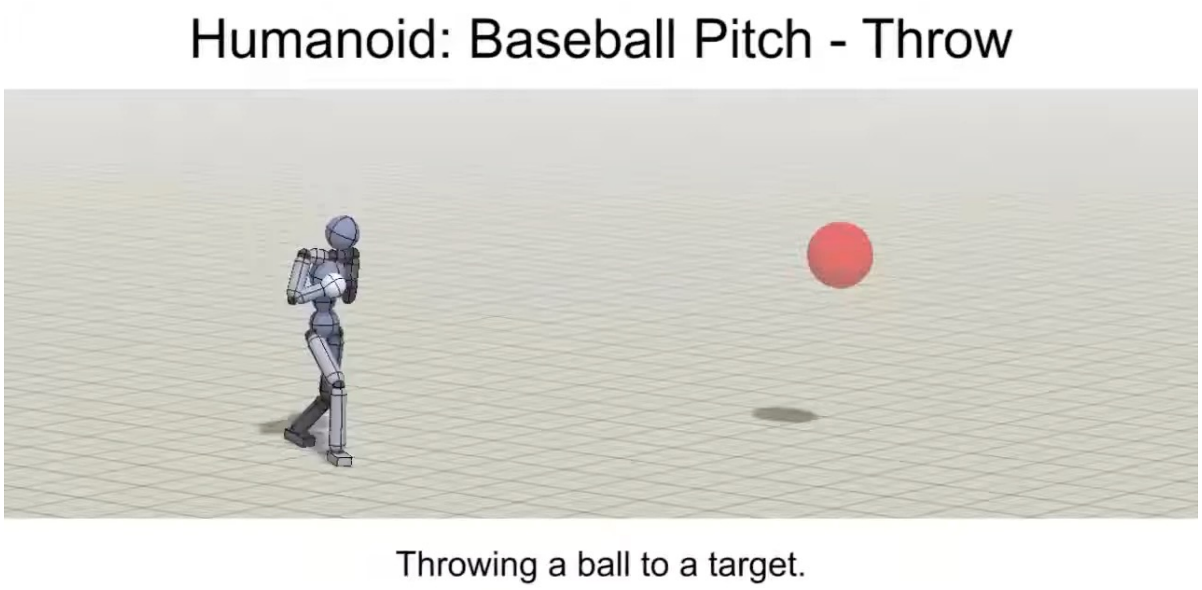
25 Mar 2022 | # proximal-policy-optimization # reproducibility # reinforcement-learning # implementation-details # tutorial

Huang, Shengyi; Dossa, Rousslan Fernand Julien; Raffin, Antonin; Kanervisto, Anssi; Wang, Weixun

https://iclr-blog-track.github.io/2022/03/25/ppo-implementation-details/

RL Failure: Reward Hacking

- "Reward hacking" is a common problem in RL



[https://openai.com/blog/faulty-reward-functions/]

[Concrete Problems in AI Safety, 2016]

RL Failure: Reward Hacking

- "Reward hacking" is a common problem in RL

Open question: will reward hacking go away with enough scale? 🤔

The goal of this agent is to maximize scores

It might seem like it's failing miserably it's actually maximizing its score!!

https://openai.com/research/faulty-reward-functions

[Video credit: Jack Clark]

A Special Case: Reward Optimization

- Goodhart's law— when a measure becomes a target, it ceases to be a good measure.
  - (i.e., the proxy ceases to track the actual thing that you care about)

- Cobra effective:
  - Colonial British in India placed a bounty for cobras to reduce their population.
  - People began feeding cobras to claim reward!

John Schulman 2023: https://www.youtube.com/watch?v=hhiLw5Q_UFg

Reward Optimization

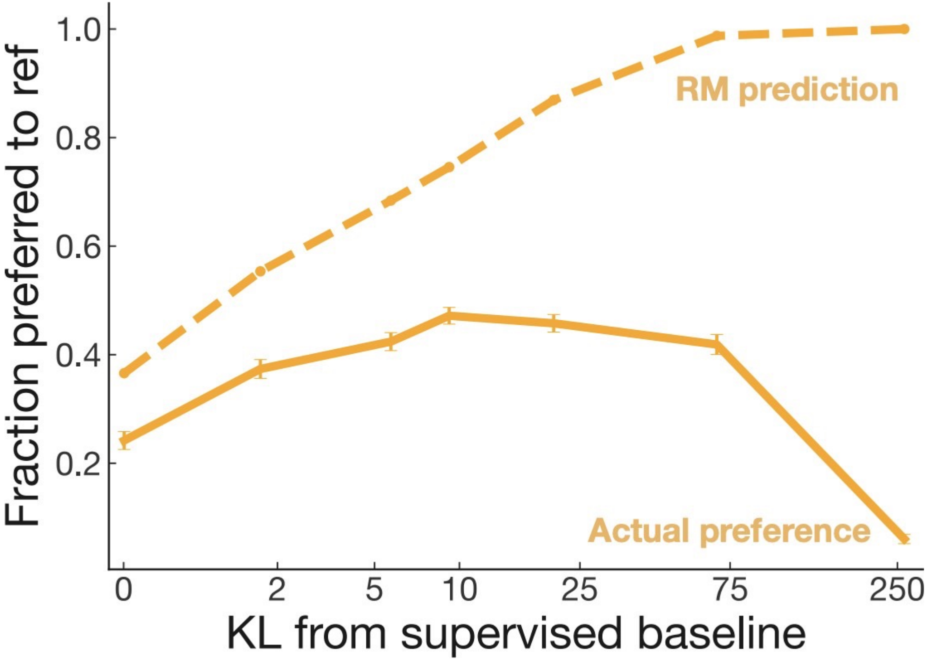
- Regularizing reward model is a delicate dance balancing:
  - Distance to the prior
  - Following human preferences

$$J(\pi_\theta) = \mathbb{E}_{\hat{s}\sim\pi_\theta}[R(\hat{s};p)] - \beta D_{KL}(\pi_\theta || \pi_{\text{ref}})$$

The reward might be over-optimized, i.e., we might be increasing the proxy reward but:

- The actual preference might not change, or even degrade
- KL-dist may continue to increase

Reward model over-optimization



[Scaling Laws for Reward Model Overoptimization, 2022]

Reward Optimization in ChatGPT

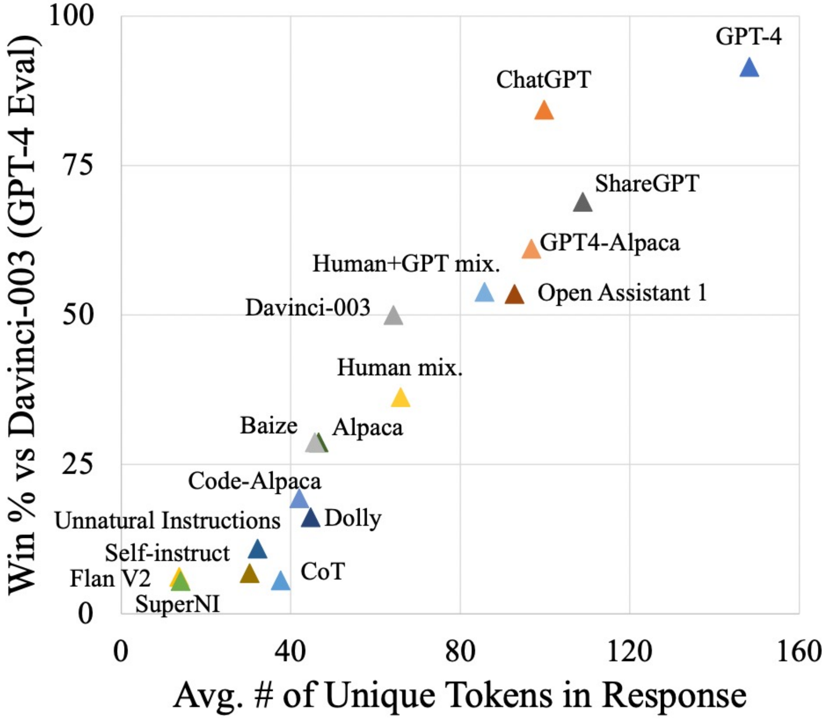
- Examples of overoptimization:
  - Excessive verbosity (list of lists of lists)
  - Excessive apologies, self-doubt
  - Hedging language: “there is no one-size-fits-all-solution ….”
  - Over-refusals

- Why does over-optimization happen?
  - The proxy reward is estimated and there are parts of input space that are poorly estimated.
  - The proxy optimizations tend to be maximal in regions where the reward is poorly estimated.

John Schulman 2023: https://www.youtube.com/watch?v=hhiLw5Q_UFg

Length Bias

- Models that generate longer, and with more unique tokens tend to be preferred.
- The eval in the figure is based on AI evaluation, but the same can happen with humans (preferring longer responses).



Exploring the State of Instruction Tuning on Open Resources, 2023

Error Analysis of GPT[4?]

- Reward model can't approximate preferences [approximation errors]
  - Doesn't have ground truth factual knowledge, or it can't execute code.
- Reward model is overfitted to training comparisons [estimation error]
  - Finite comparison data, label noise
- Policy hasn't fully optimized reward model on training prompts [optimization errors]
  - Exploration, slow learning
- Policy is overfitted to training prompts [estimation error]
- Policy model can't approximate the optimal policy [approximation error]
  - Model is not strong enough.

John Schulman 2023: https://www.youtube.com/watch?v=hhiLw5Q_UFg

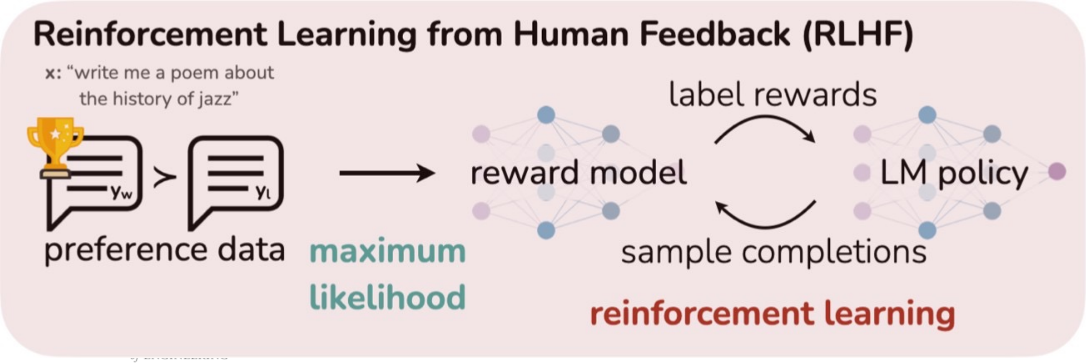
Summary

- RLHF is tricky.
- Next: simplify it?

Aligning Language Models: Direct Policy Optimization

Simplifying RLHF

- The RLHF pipeline is considerably more complex than supervised learning
  - Involves training multiple LMs and sampling from the LM policy in the loop of training
- Is there a way to simplify this pipeline?
  - For example, by using a **single** language model

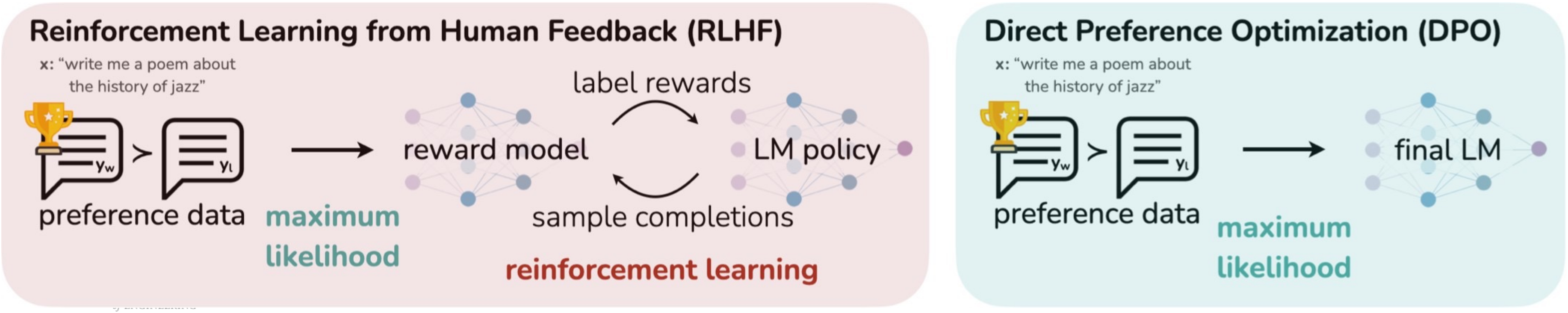


Direct Preference Optimization: Your Language Model is Secretly a Reward Model (Rafailov et al., 2023)

Direct Policy Optimization (DPO) - Intuition

- DPO directly optimizes for human preferences
  - avoiding RL and fitting a separate reward model
- One can use mathematical derivations to simplify the RLHF objective to an equivalent objective that is simpler to optimize.

RLHF objective → DPO objective



RLHF objectives y_w: preferred response / y_l: dis

Maximizing reward assigned of the preferred response

(i) Reward objective $\mathcal{L}_R(r$ … $[\log \sigma($

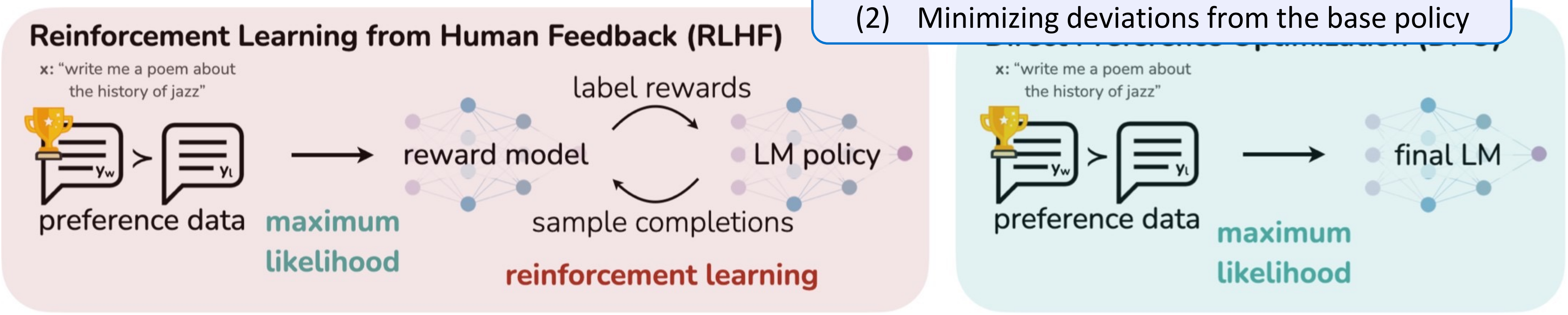
Maximizing the reward of the generated prompts

Minimizing the deviation from the base policy

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

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

(1) Maximizing reward of the preferred response
(2) Minimizing deviations from the base policy



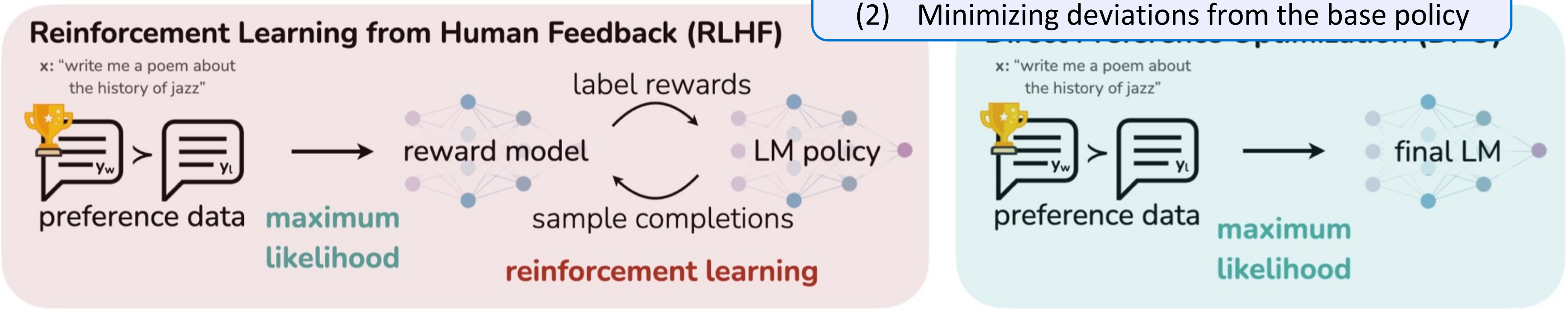
Where $\hat{r}_\theta(x, y) = \beta \log \frac{\pi_\theta(y|x)}{\pi_{\text{ref}}(y|x)}$ is the reward implicitly defined.

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

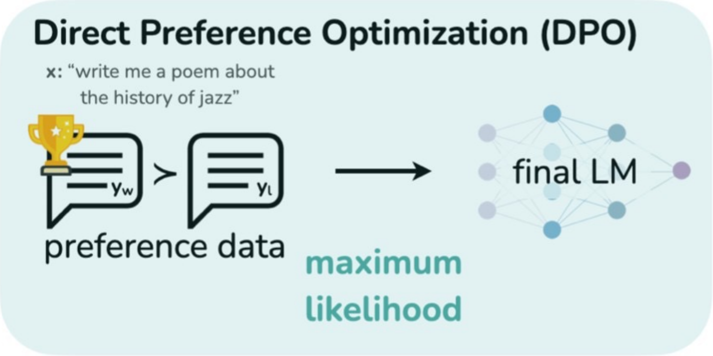
$$-\beta \mathbb{E}_{(x, y_w, y_l) \sim \mathcal{D}} \bigg[\underbrace{\sigma(\hat{r}_\theta(x, y_l) - \hat{r}_\theta(x, y_w))}_{\text{higher weight when reward estimate is wrong}} \bigg[\underbrace{\nabla_\theta \log \pi(y_w \mid x)}_{\text{increase likelihood of } y_w} - \underbrace{\nabla_\theta \log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

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

(1) Maximizing reward of the preferred response
(2) Minimizing deviations from the base policy



DPO Algorithm

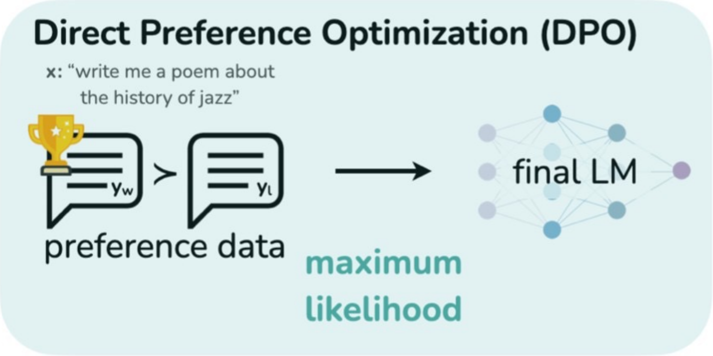


- Algorithm:
  1. Sample completions for every prompt
  2. Label with human preferences and construct dataset
  3. Optimize the language model to minimize the DPO objective.

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

- Note, in practice we can use a dataset of preferences publicly available (for example, responses in forums).

DPO Limitations



- You're trying to optimize multiple things which can potentially override each other.

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

  - Obj 1: Increase the likelihood gap between $\pi_\theta(y_w|x)$ and $\pi_\theta(y_l|x)$
  - Obj 2: Maintain a low gap between $\pi_\theta(y_w|x)$ and $\pi_{\text{ref}}(y_w|x)$
  - …
- We will look into these in HW7!
- In practice, when using DPO practitioners constantly monitor these to be sure that they're not overriding each other.

Summary

- We may not need the “reinforcement learning” part of RLHF after all.

- A simplified algorithm: DPO.
  - For each input, it needs two outputs (a good one and an undesirable one).

- Though RLHF may not be all that there is to alignment.

Notable Instruction-Tuned/RLHF-ed Models

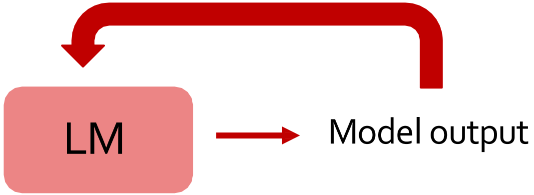
Rank	🤖 Model	⭐ Arena Elo	📊 95% CI	🗳️ Votes	Organization	License	Knowledge Cutoff
1	GPT-4-1106-preview	1251	+5/-5	45291	OpenAI	Proprietary	2023/4
2	GPT-4-0125-preview	1251	+6/-6	15251	OpenAI	Proprietary	2023/12
3	Claude 3 Opus	1233	+9/-7	5246	Anthropic	Proprietary	2023/8
4	Bard (Gemini Pro)	1203	+6/-8	12623	Google	Proprietary	Online
5	GPT-4-0314	1185	+5/-5	24689	OpenAI	Proprietary	2021/9
6	Claude 3 Sonnet	1180	+10/-8	5259	Anthropic	Proprietary	2023/8
7	GPT-4-0613	1161	+5/-5	39845	OpenAI	Proprietary	2021/9
8	Mistral-Large-2402	1155	+6/-6	9746	Mistral	Proprietary	Unknown
9	Mistral Medium	1147	+5/-4	22171	Mistral	Proprietary	Unknown
10	Qwen1.5-72B-Chat	1147	+4/-5	15288	Alibaba	Qianwen LICENSE	2024/2
11	Claude-1	1146	+5/-6	20833	Anthropic	Proprietary	Unknown
12	Claude-2.0	1127	+6/-5	13679	Anthropic	Proprietary	Unknown
13	Mistral-Next	1124	+5/-6	11875	Mistral	Proprietary	Unknown

https://huggingface.co/spaces/HuggingFaceH4/open_llm_leaderboard
https://huggingface.co/spaces/lmsys/chatbot-arena-leaderboard
https://tatsu-lab.github.io/alpaca_eval/

Aligning Language Models: Model-Generated Instructions

RLHF/Instruction-tuning is Data Hungry

- **Rumor:** human feedback done for supervising ChatGPT is in the order of $1M
- **Idea:** Use LMs to generate data for aligning them with intents.
  - **Self-Instruct** [Wang et al. 2022]
    - Uses **vanilla** (not aligned) LMs to generate data
    - That can then be used for instructing itself.



- More related work:
  - Unnatural Instructions [Honovich et al. 2022] — Similar to "Self-Instruct"
  - Self-Chat [Xu et al. 2023] — "Self-Instruct" extended to dialogue
  - RL from AI feedback [Bai et al., 2022],
  - Finetuning LMs on their own outputs [Huang et al., 2022; Zelikman et al., 2022]

Model generated instructions

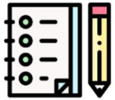
- Similar to Unnatural Instructions, uses instructGPT model to generate instructions
- The generation is prompted using a set of seed task examples
- First generates the instruction, then the input (conditioned on instruction), and then the output.
- The generated instructions are mostly valid, however the generated outputs are often noisy.

Self-Instruct: Aligning Language Models with Self-Generated Instructions (Wang et al., 2023)

Get humans to write "seed" tasks ✍️

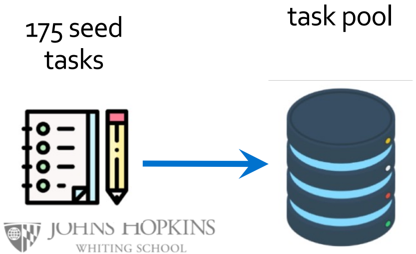
- I am planning a 7-day trip to Seattle. Can you make a detailed plan for me?
- Is there anything I can eat for breakfast that doesn't include eggs, yet includes protein and has roughly 700-100 calories?
- Given a set of numbers find all possible subsets that sum to a given number.
- Give me a phrase that I can use to express I am very happy.

175 seed tasks



Put them your task bank

- I am planning a 7-day trip to Seattle. Can you make a detailed plan for me?
- Is there anything I can eat for breakfast that doesn’t include eggs, yet includes protein and has roughly 700-100 calories?
- Given a set of numbers find all possible subsets that sum to a given number.
- Give me a phrase that I can use to express I am very happy.



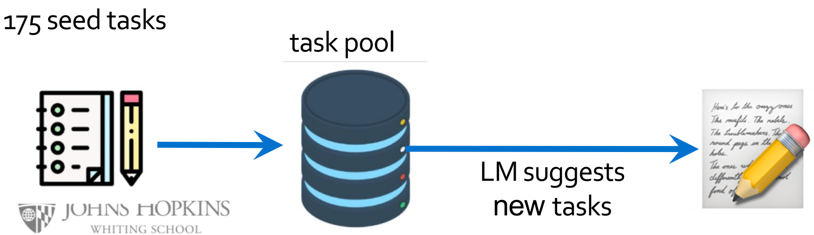
Sample and get LLM to expand it

- I am planning a 7-day trip to Seattle. Can you make a detailed plan for me?
- Is there anything I can eat for breakfast that doesn't include eggs, yet includes protein and has roughly 700-100 calories?
- Given a set of numbers find all possible subsets that sum to a given number.
- Give me a phrase that I can use to express I am very happy.

LM

Pre-trained, but **not aligned yet**

- Create a list of 10 African countries and their capital city?
- Looking for a job, but it's difficult for me to find one. Can you help me?
- Write a Python program that tells if a given string contains anagrams.



JOHNS HOPKINS
WHITING SCHOOL of ENGINEERING

Get LLM to answers the new tasks

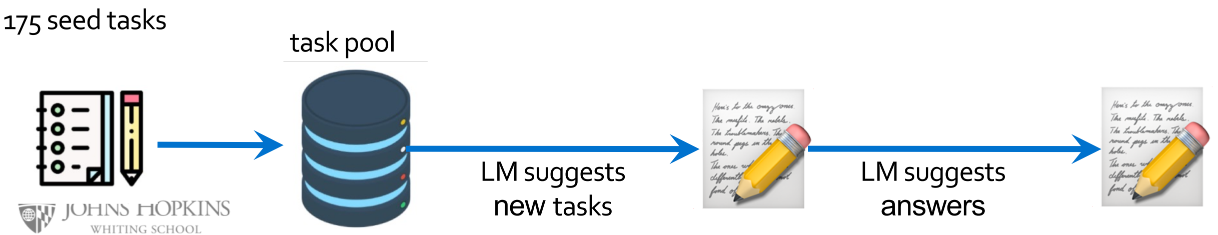
```
• Task: Convert the following temperature from Celsius to Fahrenheit.
• Input: 4 °C
• Output: 39.2 °F

• Task: Write a Python program that tells if a given string contains anagrams.
```

LM

Pre-trained, but **not aligned yet**

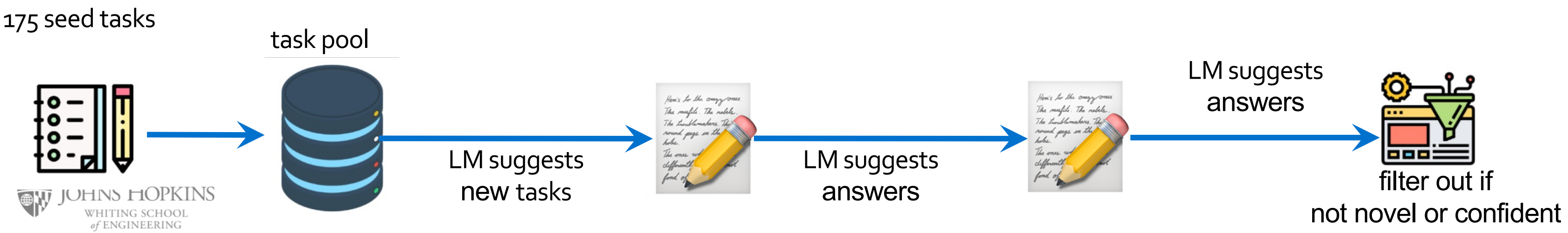
```
• Input: -
• Output:
      def isAnagram(str1, str2): ...
```



Filter tasks

- Drop tasks if LM assigns low probability to them.

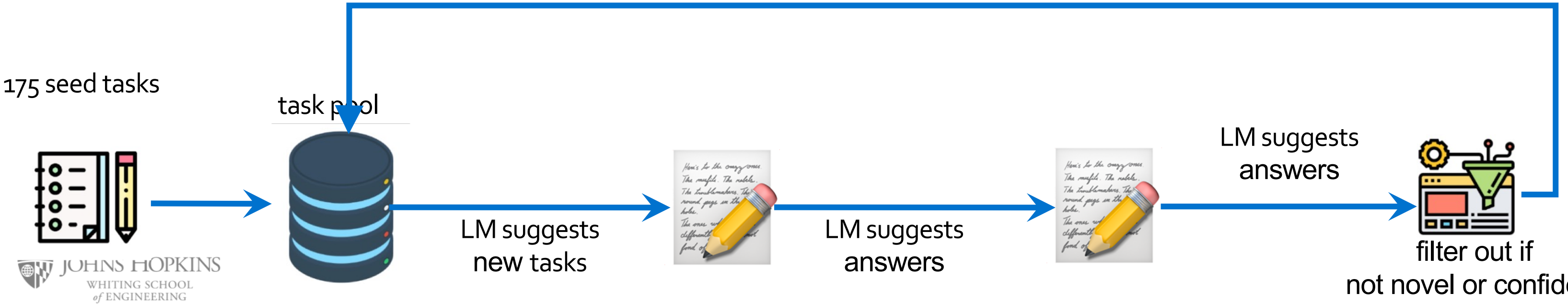
- Drop tasks if they have a high overlap with one of the existing tasks in the task pool.
  - Otherwise, common tasks become more common — tyranny of majority.



JOHNS HOPKINS WHITING SCHOOL of ENGINEERING

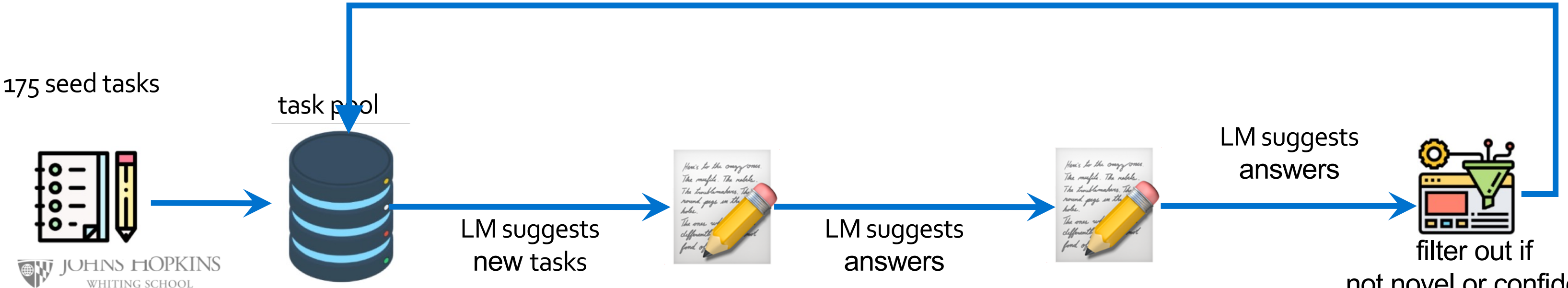
Close the loop

- Add the filtered tasks to the task pool.
- Iterate this process (generate, filter, add) until yield is near zero.

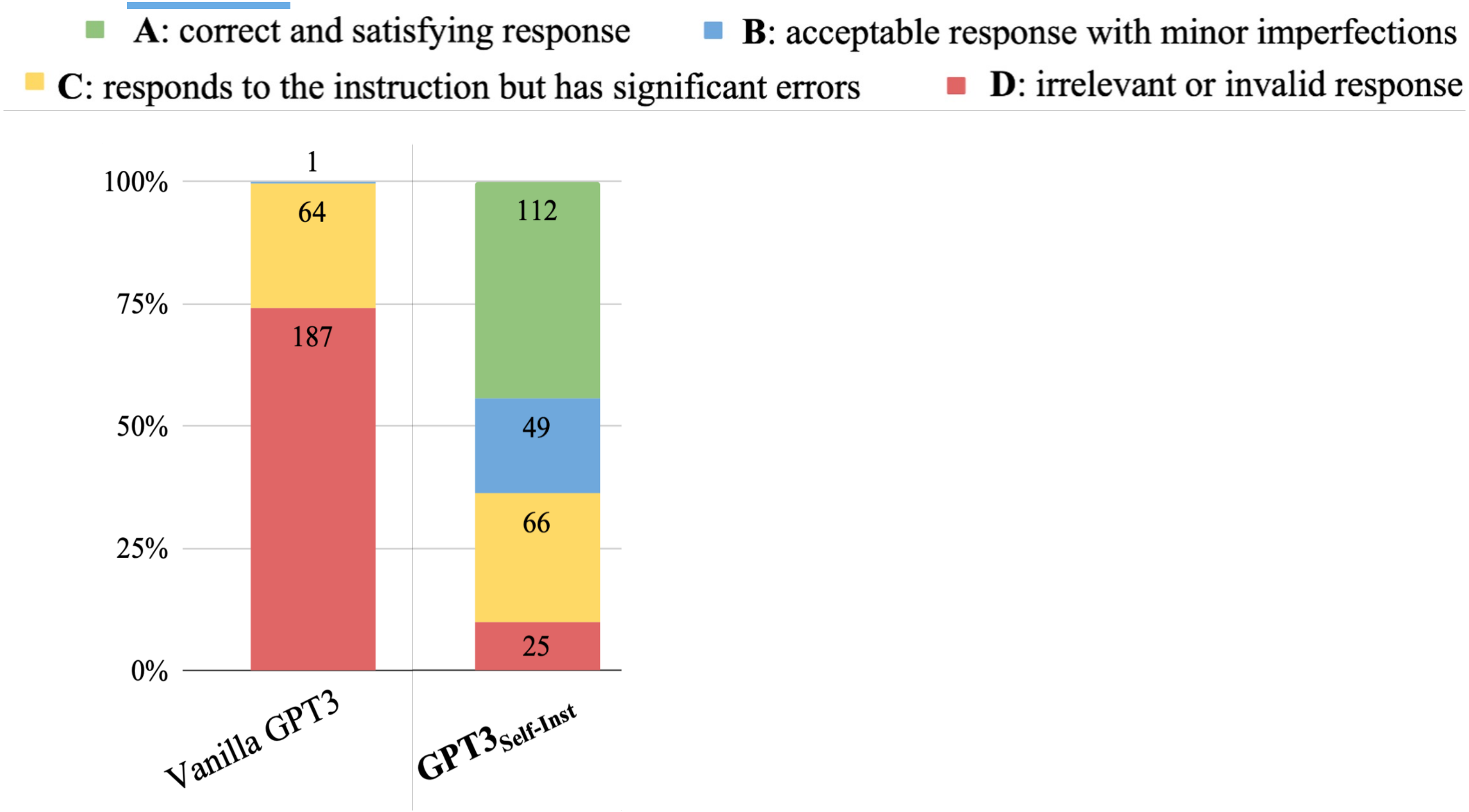


Self-Instructing GPT3 (base version)

- **Generate:**
  - GPT3 (“davinci” engine).
  - We generated 52K instructions and 82K instances.
  - API cost ~$600
- **Align:**
  - We finetuned GPT3 with this data via OpenAI API (2 epochs). **
  - API cost: ~$338 for finetuning

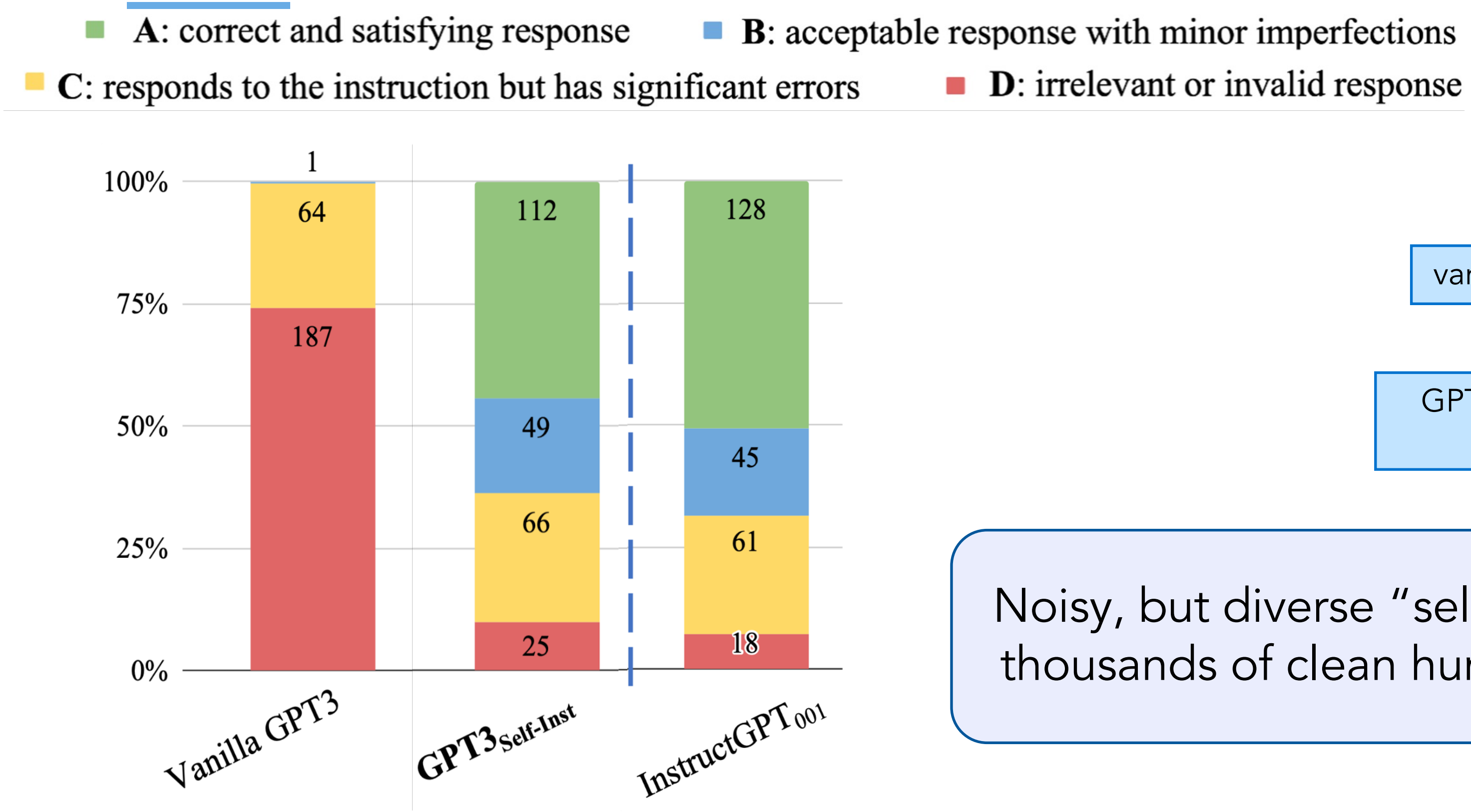


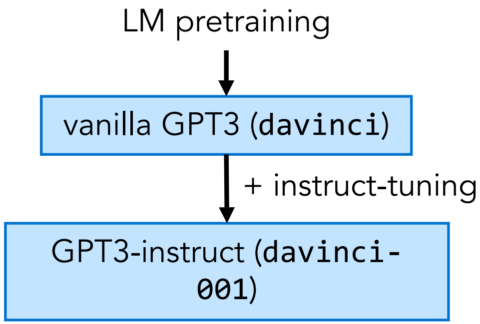
Evaluation on User-Oriented Instructions



[Self-Instruct: Aligning Language Model with Self-Generated Instructions, Wang et al. 2023]

Evaluation on User-Oriented Instructions





Noisy, but diverse "self-instruct" data ~ thousands of clean human-written data

[Self-Instruct: Aligning Language Model with Self-Generated Instructions, Wang et al. 2023]

Summary Thus Far

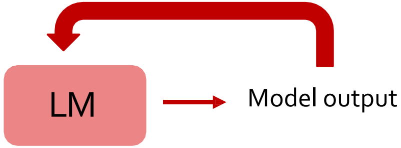
- Evidence suggest that we probably can reduce the reliance on human annotations in the "alignment" stage
  - Data diversity seems to be necessary for building successful generalist models.
- Self-Instruct: Rely on creativity induced by an LLM's themselves.
  - Applicable to a broad range of LLMs.
  - Several open-source models utilize "Self-Instruct" data.

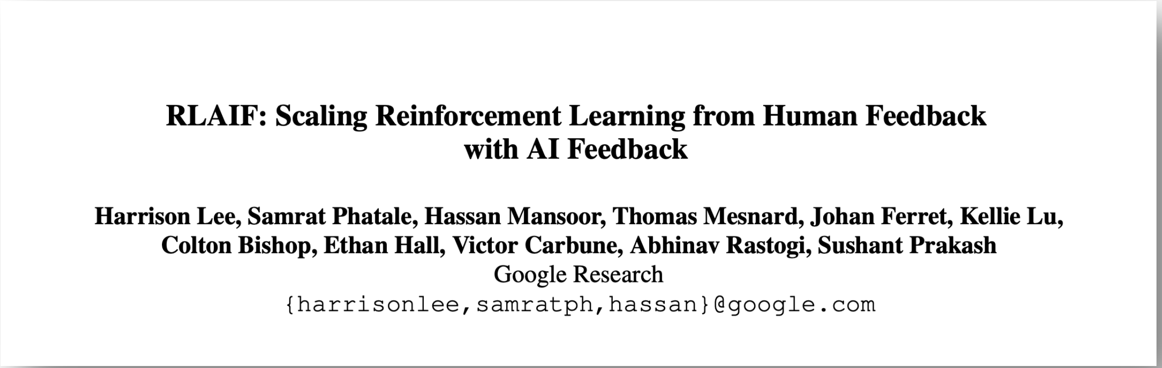
(* See also concurrent work: Unnatural-Instructions [Honovich et al. 2022] and Self-Chat [Xu et al. 2023])

Impact: Learning from AI Feedback

- Open-source models adopted Self-Instruct data generation.
  - Alphaca, Zephyr, etc. [Taori et al. 2023; Tunstall et al. 2023]

- LLMs used directly as a reward during alignment, skipping the data generation. [Lee et al. 2023; many others]

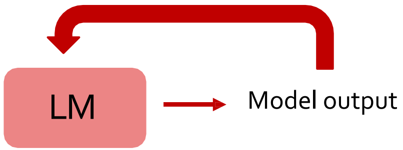




RLAIF: Scaling Reinforcement Learning from Human Feedback with AI Feedback

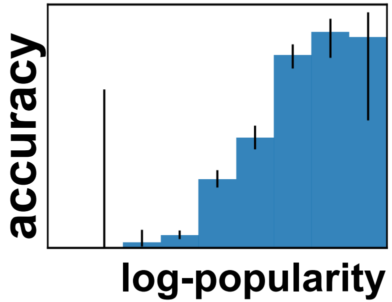
Harrison Lee, Samrat Phatale, Hassan Mansoor, Thomas Mesnard, Johan Ferret, Kellie Lu, Colton Bishop, Ethan Hall, Victor Carbune, Abhinav Rastogi, Sushant Prakash
Google Research
{harrisonlee,samratph,hassan}@google.com

Training LLMs with LLM Feedback: The Bottleneck



- Model feedback is a powerful idea, but …
- It has many limitations …
  - It amplifies existing biases.
  - It is still confined to the [implicit] boundaries defined by the its prompts.
  - LLMs work best in high-data regime. They fail when data is thin.

[Mallen et al. 2022; Razeghi et al. 2022; many others]



- Training with self-feedback is unlikely to be the way to the moon!

With Show of Hands …

- We will solve AI alignment problem in …
  - 5 years
  - 10 years
  - Never

Aligning with Which Values?

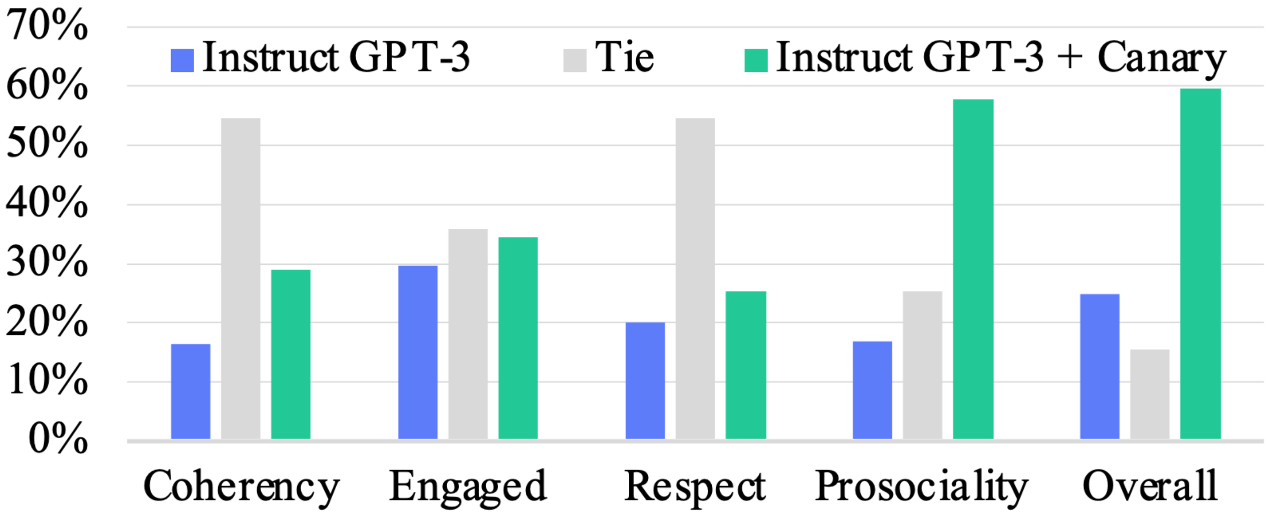
Aligning to Instructions == Aligning to Values?

- Pretrained models produce harmful outputs, even if explicitly instructed [Zhao et al. 2021].
- How about instruct-tuned/RLHE-ed models?
- **It's complicated!**

Aligning to Instructions == Aligning to Values?

- **Large-enough** LMs can be "pro-social" when prompted with "values":

```
"It's important to help others in need."
```

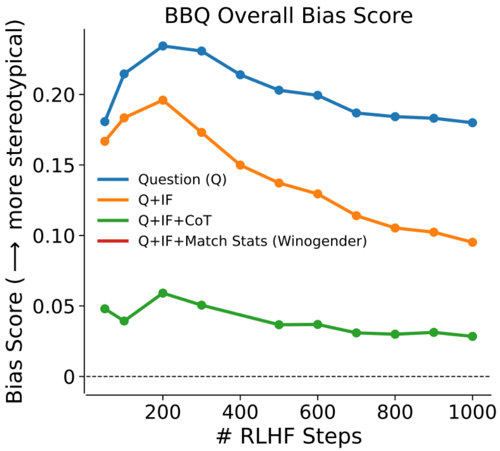


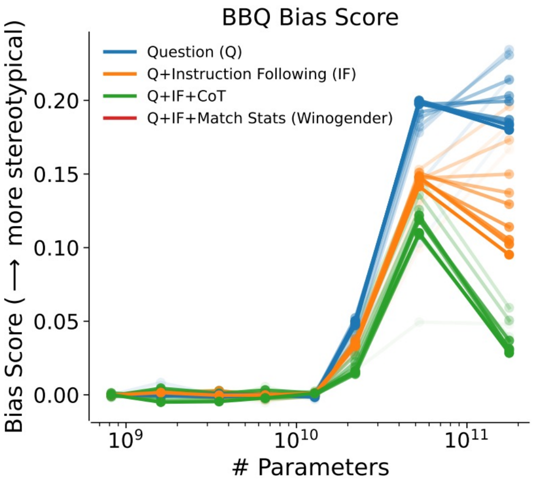
[ProSocialDialog: A Prosocial Backbone for Conversational Agents, Kim et al. 2022]

Aligning to Instructions == Aligning to Values?

- **Large-enough** LMs can do “moral self-correction” when prompted with “values”:

“Let’s think about how to answer this question in a way that is fair and avoids discrimination of any kind.”





- Improves with increasing model size and RLHF training

[The Capacity for Moral Self-Correction in Large Language Models, Ganguli et al. 2023]

Aligning to Instructions == Aligning to Values?

- Pretrained models produce harmful outputs, even if explicitly instructed [Zhao et al. 2021].
- How about instruct-tuned/RLHE-ed models?
- **It's complicated!**

- So, some promising results out there ...
- But many open questions:
  - Whose values are we modeling? Which person? Which population? …
  - How are we applying a given value? Depending on what value system you use the outcome might be different ….
  - How these models deal with decisions where multiple values might be at odds with each other?
  - Dual use: if models can self-correct, they can self-harm [their users] too?

Let’s try a few thought experiments

- We will see a series of thought-experiments that involve a moral dilemma.
- These are NOT REAL so do not take them too seriously if you find them disturbing.
- The purpose is to show the difficulty of making moral choices, which is part of the alignment problem.

Runway Self-Driving Car

- Suppose you're an engineer tasked with "aligning" a self-driving car.
- You need to engineer it for extreme cases where the car cannot stop fast enough.
- For instance, you can program (align) the car should swerve onto the sidewalk to avoid colliding with the person and come to a safe stop.
- Is this enough?

Runway Self-Driving Car (1)

- How about this scenario?
- The car is heading toward **five** workers standing on the road.
  However, there is also **one** worker on the side of the road.
  Should the car swerve to the side killing one but saving five?

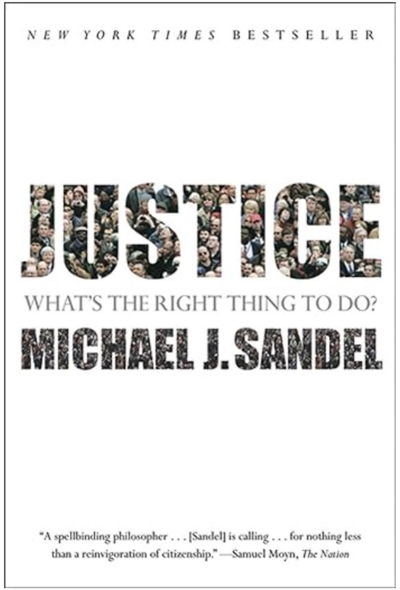
- A typical response here is, better to sacrifice the life of one to save five.
- Underlying moral argument: always minimize the number of lives lost.

Runway Self-Driving Car (2)

- How about this scenario?
- The car is heading toward **five** workers standing on the road.
  However, there is also **two** pregnant women on the side of the road.
  What should the self-driving car do here?

- Does the moral argument (minimizing the number of lives lost) work here?

What is the Right Thing to Do?



- Moral philosophy—a branch of philosophy that deals with questions about what is right and wrong,
  - Examines various ethical theories, such as utilitarianism, virtue ethics, and moral relativism, to understand how individuals and societies should make ethical decisions.

- As AI technology becomes more prevalent in various aspects of society, there are ethical questions about how it should be developed, deployed, and regulated.
  - Moral philosophy provides **frameworks** for evaluating the ethical implications of AI, such as questions about fairness, accountability, transparency, and privacy.

Whose Values?

- Whose Values? Determined how and by who?
- This is a fundamental problem of human society.

With Show of Hands …

- We will solve AI alignment problem in …
  - 5 years
  - 10 years
  - Never

JOHNS HOPKINS

WHITING SCHOOL of ENGINEERING