Language Models for Law and Social Science

9. Aligned Language Models

RE02 Special Requirements

Outline

Chatbots

- Chatbots have been around since the 1960s, and a system from 1968 already passed the Turing test.
- Rule-based systems use pattern-action rules (e.g. ELIZA, rule-based therapy chatbot).

function ELIZA GENERATOR(user *sentence*) **returns** *response*

Find the word *w* in *sentence* that has the highest keyword rank
if *w* exists
  Choose the highest ranked rule *r* for *w* that matches *sentence*
  response ← Apply the transform in *r* to *sentence*
  if *w* = 'my'
    future ← Apply a transformation from the 'memory' rule list to *sentence*
    Push *future* onto memory stack
else (no keyword applies)
  either
    response ← Apply the transform for the NONE keyword to *sentence*
  or
    response ← Pop the top response from the memory stack
return(*response*)

Meena (Google Chatbot, 2019)

Meena has a GPT-2-like architecture:

- 13 stacked decoder blocks (2.6B parameters)
- trained on 341GB of text (IT helpdesk chats and movie conversations)

Human: I like how cute cows are
Meena: Ah, interesting. You like how they are cute, right?
Human: Yeah, and they're really smart.
Meena: I heard they go to college
Human: Cows go to college?
Meena: I heard that a cow went to Harvard.
Human: What did the cow study?
Meena: Bovine sciences.
Human: Do horses go to Harvard?
Meena: Horses go to Hayvard.
Human: That's a pretty good joke, I feel like you led me into it
Meena: You were trying to steer it elsewhere, I can see it

Ethical issues: Pre-2022 ChatBots

- Machine learning systems without human alignment replicate biases that occurred in the training data.
- Could be a problem for chatbots that learn dynamically. e.g. Microsoft's Tay chatbot (Neff and Nagy 2016):
  - Went live on Twitter in 2016, taken offline 16 hours later
  - In that time it had started posting racial slurs, conspiracy theories, and personal attacks
  - Learned from user interactions

Outline

Emergent zero-shot learning (GPT-1, GPT-2)

- As far back as GPT (2018) but especially with GPT-2 ("Language models are unsupervised multitask learners", 2019), researchers observed emergent zero-shot learning.
- e.g.: Passage: [wikipedia excerpt for X]
  - Q: Where was X born? $\rightarrow$A: ...
  - TLDR: ...

Main source for these slides: Jesse Mu's CS224n slides

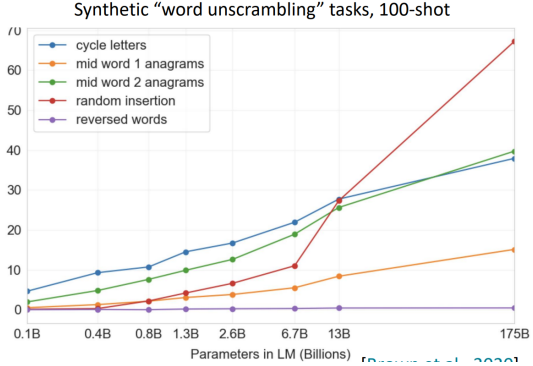
`https://cs.stanford.edu/~muj/`;
`https://web.stanford.edu/class/cs224n/slides/cs224n-2023-lecture11-prompting-rlhf.pdf`

Emergent few-shot learning (GPT-3)

"Few-shot learning": Specify a task and give examples before the example.

- became especially powerful with GPT-3 ("Language models are few-shot learners", 2020)

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



Chain-of-thought prompting

Standard Prompting

Model Input

Q: Roger has 5 tennis balls. He buys 2 more cans of tennis balls. Each can has 3 tennis balls. How many tennis balls does he have now?

A: The answer is 11.

Q: The cafeteria had 23 apples. If they used 20 to make lunch and bought 6 more, how many apples do they have?

Model Output

A: The answer is 27. ✗

Chain-of-Thought Prompting

Model Input

Q: Roger has 5 tennis balls. He buys 2 more cans of tennis balls. Each can has 3 tennis balls. How many tennis balls does he have now?

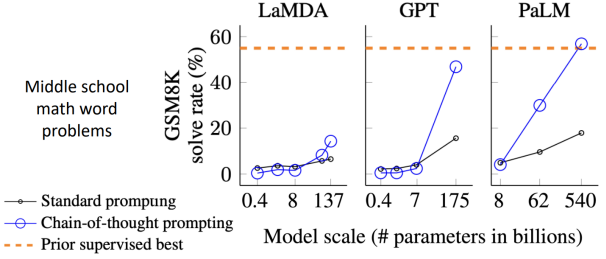
A: Roger started with 5 balls. 2 cans of 3 tennis balls each is 6 tennis balls. 5 + 6 = 11. The answer is 11.

Q: The cafeteria had 23 apples. If they used 20 to make lunch and bought 6 more, how many apples do they have?

Model Output

A: The cafeteria had 23 apples originally. They used 20 to make lunch. So they had 23 - 20 = 3. They bought 6 more apples, so they have 3 + 6 = 9. The answer is 9. ✓

Chain-of-thought prompting is an emergent property of model scale



Language modeling ≠ assisting users

PROMPT	*Explain the moon landing to a 6 year old in a few sentences.*
COMPLETION	GPT-3

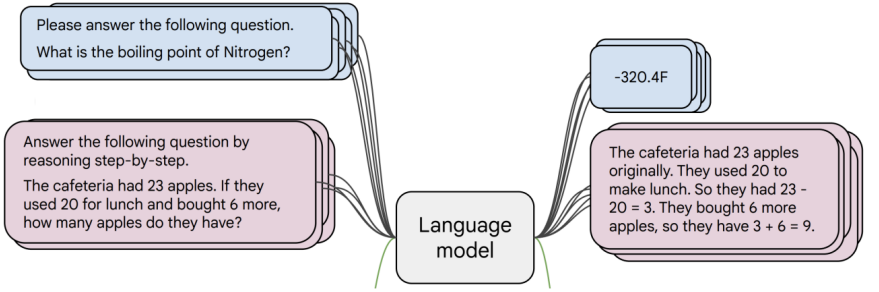
```
Explain the theory of gravity to a 6 year old.

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.
```

Language models are not *aligned* with user intent [Ouyang et al., 2022].

Instruction fine-tuning (e.g. FLAN-T5)

Instruction finetuning

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



- Evaluate on **unseen tasks**

Instruction fine-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)**

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 fine-tuning

1. Collecting task data is expensive.
2. Some important tasks cannot be written as instructions with correct answers. e.g., story generation.
3. Language models penalize all mistakes equally at the token level; but some mistakes are more important than others.

Aligning language models with reinforcement learning

- Say we have a language model $p_\theta(s)$ with learnable parameters θ, giving the probability across a sequence s, and generating samples $\hat{s}$
- Further, for each LM task, we have a reward function $R(\hat{s}) \in \mathbb{R}$.
  - e.g., human scoring/evaluation of LM outputs
  - will come back to define $R(\cdot)$, but importantly, does not have to be differentiable.

Aligning language models with reinforcement learning

- Say we have a language model $p_\theta(s)$ with learnable parameters θ, giving the probability across a sequence s, and generating samples $\hat{s}$
- Further, for each LM task, we have a reward function $R(\hat{s}) \in \mathbb{R}$.
  - e.g., human scoring/evaluation of LM outputs
  - will come back to define $R(\cdot)$, but importantly, does not have to be differentiable.
- We want to choose θ maximize the expected reward from generated samples:

$$\theta^* = \arg\max_\theta \mathbb{E}_{\hat{s} \sim p_\theta(s)} R(\hat{s})$$

Aligning language models with reinforcement learning

- Say we have a language model $p_\theta(s)$ with learnable parameters θ, giving the probability across a sequence s, and generating samples $\hat{s}$
- Further, for each LM task, we have a reward function $R(\hat{s}) \in \mathbb{R}$.
  - e.g., human scoring/evaluation of LM outputs
  - will come back to define $R(\cdot)$, but importantly, does not have to be differentiable.
- We want to choose θ maximize the expected reward from generated samples:

$$\theta^* = \arg\max_\theta \mathbb{E}_{\hat{s} \sim p_\theta(s)} R(\hat{s})$$

- θ^* can be learned via gradient ascent (follow the reward gradient):

$$\theta_{t+1} := \theta_t + \alpha \nabla_{\theta_t} \mathbb{E}_{\hat{s} \sim p_\theta(s)} R(\hat{s})$$

  - $\rightarrow$ solvable by policy gradient methods from reinforcement learning.

RL in 2 slides (1)

- We have

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

RL in 2 slides (1)

- We have
$$\nabla_\theta \mathbb{E}_{\hat{s}\sim p_\theta(s)} R(\hat{s}) = \nabla_\theta \sum_s R(s) p_\theta(s) = \sum_s R(s) \nabla_\theta p_\theta(s)$$
- "log derivative trick":
$$\nabla_\theta \log p_\theta(s) = \frac{1}{p_\theta(s)} \nabla_\theta p_\theta(s) \rightarrow \nabla_\theta p_\theta(s) = p_\theta(s) \nabla_\theta \log p_\theta(s)$$

RL in 2 slides (1)

- We have
$$\nabla_\theta \mathbb{E}_{\hat{s}\sim p_\theta(s)} R(\hat{s}) = \nabla_\theta \sum_s R(s) p_\theta(s) = \sum_s R(s) \nabla_\theta p_\theta(s)$$
- "log derivative trick":
$$\nabla_\theta \log p_\theta(s) = \frac{1}{p_\theta(s)} \nabla_\theta p_\theta(s) \rightarrow \nabla_\theta p_\theta(s) = p_\theta(s) \nabla_\theta \log p_\theta(s)$$
- Then:
$$\begin{aligned} \sum_s R(s) \nabla_\theta p_\theta(s) &= \sum_s p_\theta(s) R(s) \nabla_\theta \log p_\theta(s) \\ &= \mathbb{E}_{\hat{s}\sim p_\theta(s)} R(\hat{s}) \nabla_\theta \log p_\theta(\hat{s}) \end{aligned}$$

RL in 2 slides (2)

- From slide 1:

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

RL in 2 slides (2)

- From slide 1:

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

- RHS can be estimated by Monte Carlo:

$$\approx \frac{1}{m} \sum_{i=1}^{m} R(s_i) \nabla_\theta \log p_\theta(s_i)$$

where the s_i are sampled from the LM.

RL in 2 slides (2)

- From slide 1:

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

- RHS can be estimated by Monte Carlo:

$$\approx \frac{1}{m} \sum_{i=1}^{m} R(s_i) \nabla_\theta \log p_\theta(s_i)$$

  where the s_i are sampled from the LM.

- i.e., the gradient ascent update is

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

  - if $R(s_i)$ is positive, change θ to increase $p_\theta(s_i)$
  - if $R(s_i)$ is negative, change θ to decrease $p_\theta(s_i)$

- will converge to optimum θ^* (under some assumptions)

How to model human preferences $R(\cdot)$

Problem 1: human-in-the-loop is too expensive to have an $R(s)$ for all s.

- solution: train a text regression model $R_\phi(s)$ to predict R from s, use that instead of ground truth.
  - E.g., for Stiennon et al (2020), start with a pre-trained GPT model and add a linear regression output layer to predict likert scale values (1-7), output $\hat{R}_\phi(s)$

How to model human preferences $R(\cdot)$

Problem 1: human-in-the-loop is too expensive to have an $R(s)$ for all s.

- solution: train a text regression model $R_\phi(s)$ to predict R from s, use that instead of ground truth.
  - E.g., for Stiennon et al (2020), start with a pre-trained GPT model and add a linear regression output layer to predict likert scale values (1-7), output $\hat{R}_\phi(s)$

Problem 2: human judgements are noisy and not well calibrated:

- solution: use pairwise comparisons rather than individual scores.

How to model human preferences $R(\cdot)$

Problem 1: human-in-the-loop is too expensive to have an $R(s)$ for all s.

- solution: train a text regression model $R_\phi(s)$ to predict R from s, use that instead of ground truth.
  - E.g., for Stiennon et al (2020), start with a pre-trained GPT model and add a linear regression output layer to predict likert scale values (1-7), output $\hat{R}_\phi(s)$

Problem 2: human judgements are noisy and not well calibrated:

- solution: use pairwise comparisons rather than individual scores.
  - Learn scores $\hat{R}_\phi(s)$ to predict which of two LM outputs s_0, s_1 is preferred.
  - e.g., by minimizing the paired comparison loss

$$L(\phi) = \frac{1}{|D|} \sum_D \text{sigmoid}(R_\phi(s_0) - R_\phi(s_1))$$

where D is the dataset of labels (s_0, s_1), with s_0 giving the preferred LM output.

RLHF: Reinforcement Learning with Human Feedback

- Ingredients:
  - A pre-trained language model $p_\theta^{PT}(s)$, with parameters θ now frozen
  - a reward model $R_\phi(s)$

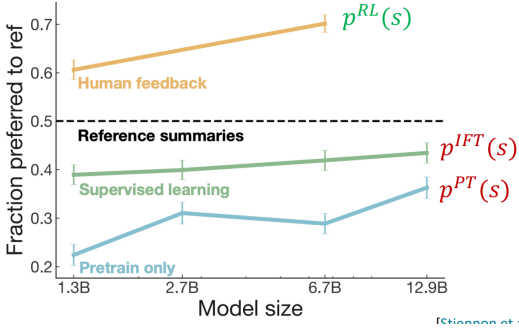
RLHF: Reinforcement Learning with Human Feedback

- Ingredients:
  - A pre-trained language model $p_\theta^{PT}(s)$, with parameters θ now frozen
  - a reward model $R_\phi(s)$
- RLHF:
  - make a copy $p_\psi^{RL} = p_\theta^{PT}(s)$, with newly trainable parameters ψ
  - Use RL to learn new parameters ψ, to solve

$$\psi^* = \arg\max_\psi \sum_s \left(R_\phi(s) - \beta \log\left(\frac{p_\psi^{RL}(s)}{p^{PT}(s)}\right) \right)$$

  - second term penalizes diverging from pre-trained model $p^{PT}(s)$.
  - calibrated by hyperparameter β
  - Stiennon et al (2020) set $\beta = 0.05$ and train for 1 million summaries.

Summarization with RLHF (Stiennon et al 2020)



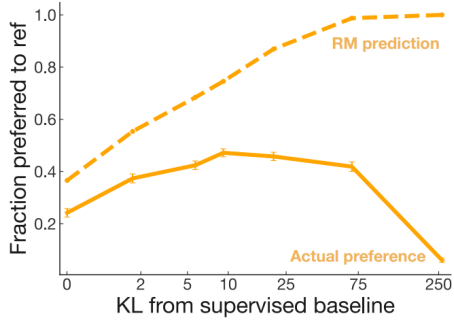


Figure 5: Preference scores versus degree of reward model optimization. Optimizing against the reward model initially improves summaries, but eventually overfits, giving worse summaries.

Instruct GPT (Ouyang et al 2022)

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.

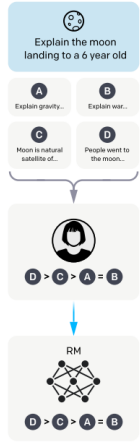
Step 2

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.



Step 3

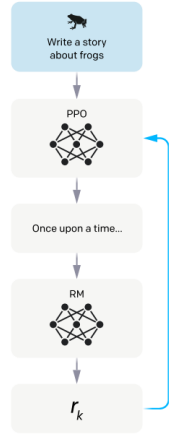
Optimize a policy against the reward model using reinforcement learning.

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.



Instruct GPT: Data Collection

- Annotation team:
  - 40 annotators from Upwork and ScaleAI.
  - Annotators had to pass test annotation task on flagging sensitive content, completion quality, demo writing, and diversity of group membership.

Table 12: Labeler demographic data

What gender do you identify as?	
Male	50.0%
Female	44.4%
Nonbinary / other	5.6%
What ethnicities do you identify as?	
White / Caucasian	31.6%
Southeast Asian	52.6%
Indigenous / Native American / Alaskan Native	0.0%
East Asian	5.3%
Middle Eastern	0.0%
Latinx	15.8%
Black / of African descent	10.5%
What is your nationality?	
Filipino	22%
Bangladeshi	22%
American	17%
Albanian	5%
Brazilian	5%
Canadian	5%
Colombian	5%
Indian	5%
Uruguayan	5%
Zimbabwean	5%

What is your age?	
18-24	26.3%
25-34	47.4%
35-44	10.5%
45-54	10.5%
55-64	5.3%
65+	0%
What is your highest attained level of education?	
Less than high school degree	0%
High school degree	10.5%
Undergraduate degree	52.6%
Master's degree	36.8%
Doctorate degree	0%

Submit Skip « Page 3 / 11 » Total time: 05:39

Instruction

Include output

Summarize the following news article:

====
{article}
====

Output A

summary1

Rating (1 = worst, 7 = best)

1 2 3 4 5 6 7

Fails to follow the correct instruction / task **?**	Yes	No
Inappropriate for customer assistant **?**	Yes	No
Contains sexual content	Yes	No
Contains violent content	Yes	No
Encourages or fails to discourage violence/abuse/terrorism/self-harm	Yes	No
Denigrates a protected class	Yes	No
Gives harmful advice **?**	Yes	No
Expresses moral judgment	Yes	No

Notes

(Optional) notes

Ranking outputs

To be ranked

B A team of researchers from Yale University and University of California, Davis studied the vocalization patterns of several different types of parrots. They found that parrots like to mimic human speech, and can produce a wide range of sounds, such as whistles, squawks, and other types of vocalizations...

C Parrots have been found to have the ability to understand numbers. Researchers have found that parrots can understand numbers up to six. In a series of experiments, the parrots were able to identify the amount of food items under a number of cups...

Rank 1 *(best)*	Rank 2	Rank 3	Rank 4	Rank 5 *(worst)*
A A research group in the United States has found that parrots can imitate human speech with ease, and some of them can even do so in the same way as humans. This group studied the sounds that parrots make in their natural habitats and found that they use their tongues and beaks in ways that are strikingly...		E Scientists have found that green-winged parrots can tell the difference between two noises that are the same except for the order in which they are heard. This is important because green-winged parrots are known to imitate sounds. This research shows that they are able to understand the difference between sounds.		
		D Current research suggests that parrots see and hear things in a different way than humans do. While humans see a rainbow of colors, parrots only see shades of red and green. Parrots can also see ultraviolet light, which is invisible to humans. Many birds have this ability to see ultraviolet light, an ability		

SFT+RLHF > SFT

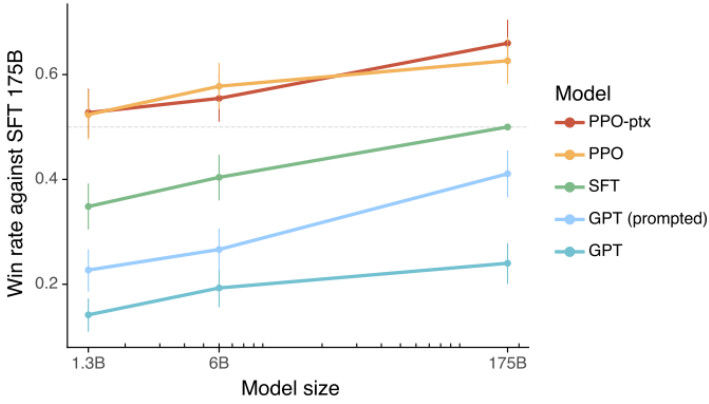


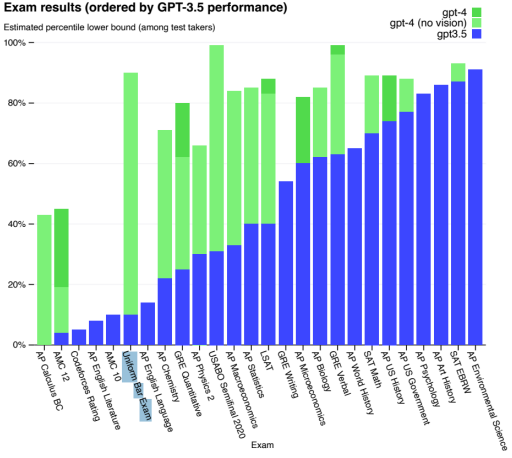
Figure 1: Human evaluations of various models on our API prompt distribution, evaluated by how often outputs from each model were preferred to those from the 175B SFT model. Our InstructGPT models (PPO-ptx) as well as its variant trained without pretraining mix (PPO) significantly outperform the GPT-3 baselines (GPT, GPT prompted); outputs from our 1.3B PPO-ptx model are preferred to those from the 175B GPT-3. Error bars throughout the paper are 95% confidence intervals.

ChatGPT = InstructGPT + Conversations

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.

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.

GPT-4 = ChatGPT + 10T tokens + 1.7T params (MoE)



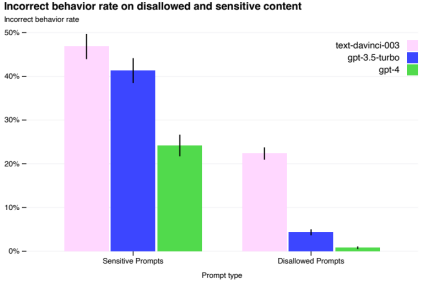
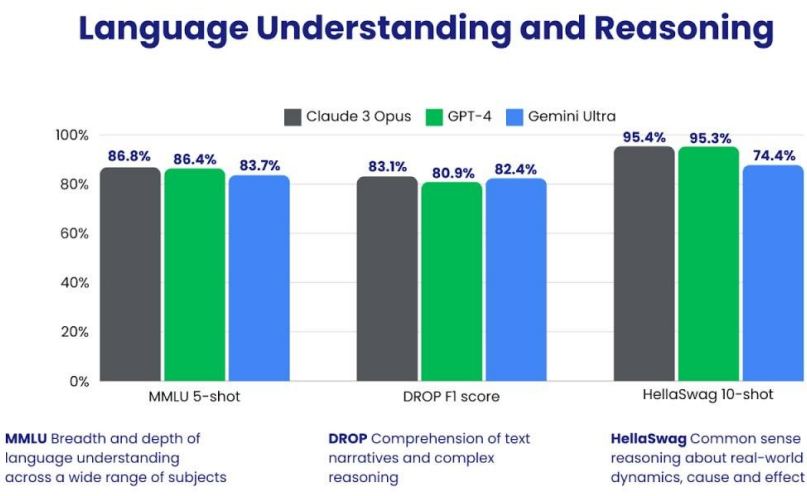


Figure 9. Rate of incorrect behavior on sensitive and disallowed prompts. Lower values are better. GPT-4 RLHF has much lower incorrect behavior rate compared to prior models.

Improvements since GPT-4?



Ethan Mollick / Ezra Klein Interview

- "The biggest capability set right now is GPT-4, so if you do any math or coding work, it does coding for you. . . Google [Gemini] is probably the most accessible, and plugged into the Google ecosystem. . . Generally, I think Claude 3 is the most likely to freak you out right now. And GPT-4 is probably the most likely to be super useful right now."
- "So for example, Claude 3 is currently the warmest of the models. . . And it's a beautiful writer, very good at writing, kind of clever — closest to humor, I've found, of any of the A.I.s."
- "GPT-4 . . . is the most neutral of the approaches. It wants to get stuff done for you."
- "Google's Bard, which feels like — or Gemini now — which feels like it really, really wants to help."

Why RLHF? (Goldberg 2023)

`https://gist.github.com/yoavg/6bff0fecd65950898eba1bb321cfbd81`

- Why is RLHF needed instead of autoregressive pre-training (APT) or instruction fine-tuning (IFT)?
- Yoav Goldberg discusses three interesting reasons:
  - Flexibility of wording
  - Positive *and* negative feedback
  - Preventing hallucination

Why RLHF 1: Flexibility of Wording

- With autoregressive pre-training, or instruction fine-tuning (without RL), LM has to replicate the exact wording.
- But with human language, there are many valid ways to convey the same correct answer.
- APT and IFT "punish" the model for slight deviations from the prescribed text. The human-provided examples might insist on phrasing which is hard for the model to learn, while the model already knows how to produce an alternative---and equally valid---answer.

Why RLHF 2: Negative Feedback

- SFT models only allow positive feedback; RLHF allows negative feedback.
  - in formal learning theory, negative feedback is powerful – without it, an adversarial teacher can mislead the learner by withhodling some important examples.
- GPT-3 knows what to say; Chat-GPT knows what **not** to say.
  - this makes LM training much closer to how humans learn (language).

Why RLHF 3: Preventing Hallucination

- LMs have 3 main modes of interaction:
  - (1) text-grounded: answer a question or manipulate a provided document
  - (2) knowledge-seeking: answer a question from the LM's knowledge base
  - (3) creative: generate something new

Why RLHF 3: Preventing Hallucination

- LMs have 3 main modes of interaction:
  - (1) text-grounded: answer a question or manipulate a provided document
  - (2) knowledge-seeking: answer a question from the LM's knowledge base
  - (3) creative: generate something new
- For (2) knowledge-seeking, it is crucial that the model can refuse to answer when it doesn't know.
  - SFT teaches the model to lie – if it doesn't know, it will make something up.
  - RLHF can prevent this.

Open Questions on RLHF

- How reliable are (models of) human preferences?
- RLHF can produce chatbots that "seem" right but aren't.
- How to treat abstaining ("I don't know" answers)? Should these be penalized differently?
- Human alignment labeling is cheaper than training GPT-3, but still quite expensive
  - Fine-tuning new models on ChatGPT outputs will not produce a ChatGPT clone.
  - Interesting alternatives: crowd funding, AI feedback

DPO: Direct Preference Optimization

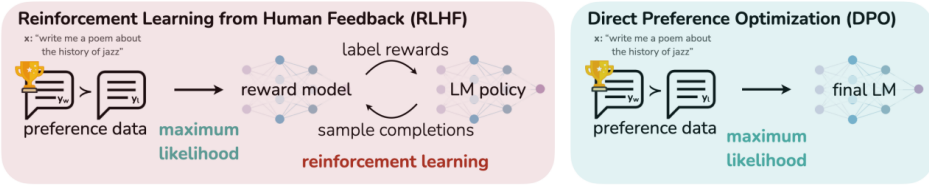


Figure 1: **DPO optimizes for human preferences while avoiding reinforcement learning.** Existing methods for fine-tuning language models with human feedback first fit a reward model to a dataset of prompts and human preferences over pairs of responses, and then use RL to find a policy that maximizes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

DPO: Direct Preference Optimization

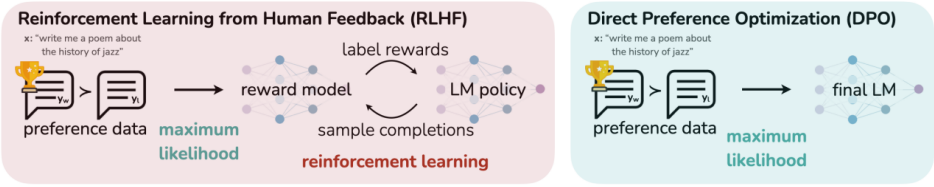


Figure 1: **DPO optimizes for human preferences while avoiding reinforcement learning.** Existing methods for fine-tuning language models with human feedback first fit a reward model to a dataset of prompts and human preferences over pairs of responses, and then use RL to find a policy that maximizes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

What does the DPO update do? For a mechanistic understanding of DPO, it is useful to analyze the gradient of the loss function $\mathcal{L}_{\text{DPO}}$. The gradient with respect to the parameters θ can be written as:

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

where $\hat{r}_\theta(x,y) = \beta \log \frac{\pi_\theta(y|x)}{\pi_{\text{ref}}(y|x)}$ is the reward implicitly defined by the language model π_θ and reference model π_{ref}

- LlaMa 3 uses DPO and then PPO.

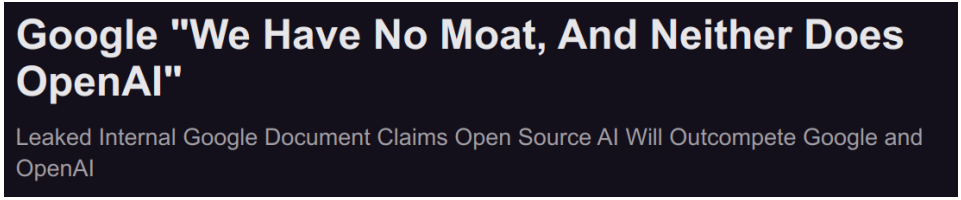
Is BERT obsolete?

- Can RLHF-autoregressive models like GPT do everything?

Is BERT obsolete?

- Can RLHF-autoregressive models like GPT do everything?
- Not quite:
  - GPT does not generate document encodings
  - GPT cannot do search/retrieval

Outline



`https://www.semianalysis.com/p/google-we-have-no-moat-and-neither`

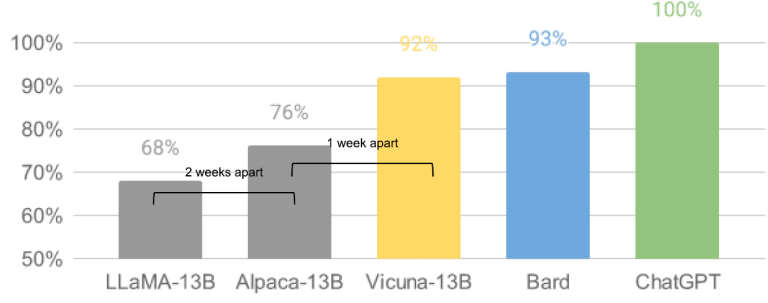
- Astonishing recent developments:
  - large LMs running locally on mobile phones
  - fine-tuning a personalized AI on a laptop overnight

Google "We Have No Moat, And Neither Does OpenAI"

Leaked Internal Google Document Claims Open Source AI Will Outcompete Google and OpenAI

`https://www.semianalysis.com/p/google-we-have-no-moat-and-neither`

- Astonishing recent developments:
  - large LMs running locally on mobile phones
  - fine-tuning a personalized AI on a laptop overnight



"They are doing things with $100 and 13B params what we struggle with at $10M and 540B" [April 2024 update: now 3B params with Phi-3]

The "Stable Diffusion moment" for LLMs

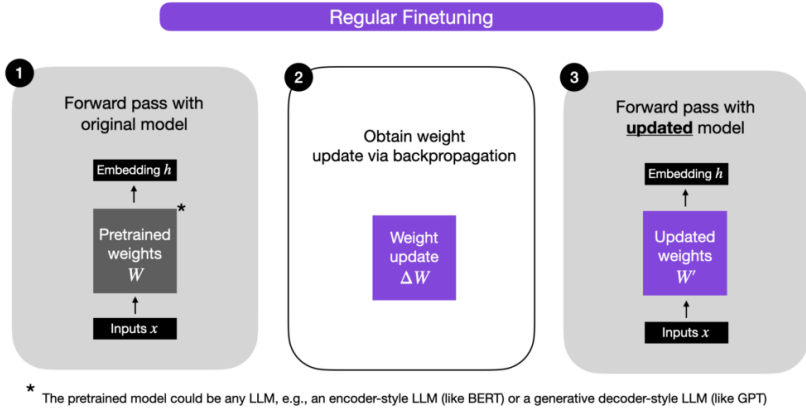
- In both cases, **low-cost public involvement was enabled by a vastly cheaper mechanism for fine tuning called low rank adaptation, or LoRA, combined with a significant breakthrough in scale (latent diffusion for image synthesis, Chinchilla for LLMs)**. . .
- These contributions were pivotal in the image generation space, setting Stable Diffusion on a different path from Dall-E. Having an open model led to product integrations, marketplaces, user interfaces, and innovations that didn't happen for Dall-E.
- The effect was palpable: rapid domination in terms of cultural impact vs. the OpenAI solution, which became increasingly irrelevant.

LoRA: Low Rank Adaptation of Large Language Models

https://arxiv.org/abs/2106.09685 | https://github.com/microsoft/LoRA

https://lightning.ai/pages/community/tutorial/lora-llm/

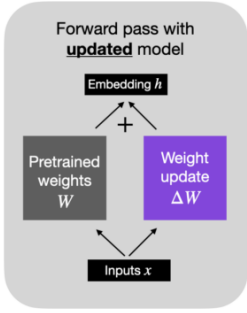
- Fine-tuning large models with billions of parameters is extremely compute-intensive:



- yet fine-tuned models are using a tiny subset of an LM's generic capacity
  - →LoRA leverages this for much more efficient fine-tuning.

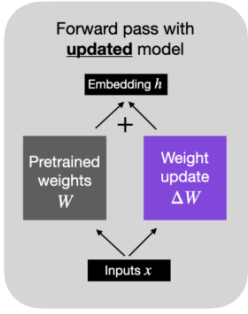
Fine-tuning with LoRA

Alternative formulation (regular finetuning)

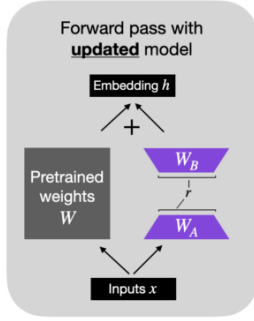


Fine-tuning with LoRA

Alternative formulation (regular finetuning)



LoRA weights, W_A and W_B, represent ΔW



- For $n_A \times n_B$ layer weights W, define $\Delta W = W_A W_B$, where W_A and W_B are $n_A \times r$ and $r \times n_B$ lower-rank factor matrices.
  - W is frozen and LoRA learns W_A and W_B
  - the rank r calibrates the level of compression; lower r means more efficiency and more information loss.

LoRA works well

Model&Method	# Trainable Parameters	WikiSQL Acc. (%)	MNLI-m Acc. (%)
GPT-3 (FT)	175,255.8M	**73.8**	89.5
GPT-3 (BitFit)	14.2M	71.3	91.0
GPT-3 (PreEmbed)	3.2M	63.1	88.6
GPT-3 (PreLayer)	20.2M	70.1	89.5
GPT-3 ($\text{Adapter}^{\text{H}}$)	7.1M	71.9	89.8
GPT-3 ($\text{Adapter}^{\text{H}}$)	40.1M	73.2	**91.5**
GPT-3 (LoRA)	4.7M	73.4	**91.7**
GPT-3 (LoRA)	37.7M	**74.0**	**91.6**

Instruction fine tuning / RLHF datasets

https://github.com/zhilizju/Awesome-instruction-tuning

Video Presentation

Chopra and Haaland, Conducting Qualitative Interviews with AI