

# **COMP0087**

## Statistical Natural Language Processing

### Lecture 9 – Post-Training

Slides present original content as well as content based on what was previously developed by:

- Jesse Mu
- Dan Goldwasser
- Maria Pacheco
- CS224n

# What does an LLM learn from pre-training?

- *London is located in \_\_\_\_\_.* [Factual Knowledge]
- *I read \_\_\_ book while I was in the library.* [syntax]
- *The cat ran away from my dog, thinking \_\_\_ was going to bite .* [coreference]
- *Overall, the value I got from the two hours watching it was the sum total of the popcorn and the drink. The movie was \_\_\_.* [sentiment]
- Elly went into the kitchen to make some tea. Standing next to Elly, Maya was eating her lunch. Elly left the \_\_\_\_\_. [some reasoning – this is harder]
- I was thinking about the sequence that goes 1, 1, 2, 3, 5, 8, 13, 21, \_\_\_\_ [some basic arithmetic; they don't learn the Fibonnaci sequence]

# Probing what is captured by pre-training

## Language Models as Knowledge Bases?

Fabio Petroni<sup>1</sup> Tim Rocktäschel<sup>1,2</sup> Patrick Lewis<sup>1,2</sup> Anton Bakhtin<sup>1</sup>  
Yuxiang Wu<sup>1,2</sup> Alexander H. Miller<sup>1</sup> Sebastian Riedel<sup>1,2</sup>

<sup>1</sup>Facebook AI Research

<sup>2</sup>University College London

Sep 2019

## Multilingual LAMA: Investigating Knowledge in Multilingual Pretrained Language Models

Feb 2021

Nora Kassner\*, Philipp Dufter\*, Hinrich Schütze

Center for Information and Language Processing (CIS), LMU Munich, Germany

## How Context Affects Language Models' Factual Predictions

Fabio Petroni<sup>1</sup>  
Patrick Lewis<sup>1,2</sup>  
Aleksandra Piktus<sup>1</sup>  
Tim Rocktäschel<sup>1,2</sup>  
Yuxiang Wu<sup>2</sup>  
Alexander H. Miller<sup>1</sup>  
Sebastian Riedel<sup>1,2</sup>  
<sup>1</sup>Facebook AI Research  
<sup>2</sup>University College London

FABIOPETRONI@FB.COM  
PLEWIS@FB.COM  
PIKTUS@FB.COM  
ROCKT@FB.COM  
YUXIANG.WU.18@UCL.AC.UK  
AHM@FB.COM  
SRIEDEL@FB.COM

May 2020

## Rewire-then-Probe: A Contrastive Recipe for Probing Biomedical Knowledge of Pre-trained Language Models

Zaiqiao Meng<sup>\*♦\*</sup> Fangyu Liu<sup>\*\*</sup> Ehsan Shareghi<sup>\*\*</sup>  
Yixuan Su<sup>\*</sup> Charlotte Collins<sup>\*</sup> Nigel Collier<sup>\*</sup>

<sup>\*</sup>Language Technology Lab, University of Cambridge

<sup>°</sup>Department of Computing Science, University of Glasgow

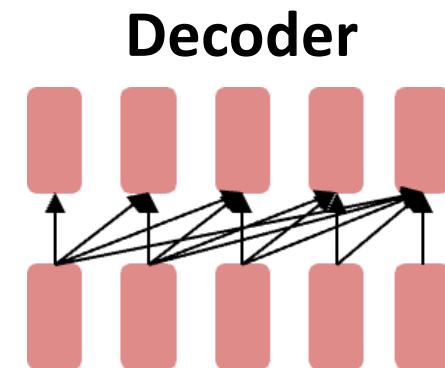
May 2022

# Emergent abilities of large language models: GPT (2018)

Let's revisit the Generative Pretrained Transformer (GPT) models from OpenAI as an example:

**GPT (117M parameters; [Radford et al., 2018](#))**

- Transformer decoder with 12 layers.
- Trained on BooksCorpus: over 7000 unique books (**4.6GB** text).



Showed that language modeling at scale can be an effective pretraining technique for downstream tasks like natural language inference.

[START] *The man is in the doorway* [DELIM] *The person is near the door* [EXTRACT]

entailment

# Emergent abilities of large language models: GPT-2 (2019)

**GPT-2 (1.5B parameters; [Radford et al., 2019](#))**

- Same architecture as GPT, just bigger (117M -> 1.5B)
- But trained on **much more data**: 4GB -> **40GB** of internet text data (WebText)
  - Scrape links posted on Reddit w/ at least 3 upvotes (rough proxy of human quality)

---

## Language Models are Unsupervised Multitask Learners

---

Alec Radford \*<sup>1</sup> Jeffrey Wu \*<sup>1</sup> Rewon Child<sup>1</sup> David Luan<sup>1</sup> Dario Amodei \*\*<sup>1</sup> Ilya Sutskever \*\*<sup>1</sup>

# Emergent zero-shot learning

One key emergent ability in GPT-2 is **zero-shot learning**: the ability to do many tasks with **no examples**, and **no gradient updates**, by simply:

- Specifying the right sequence prediction problem (e.g. question answering):

Passage: Tom Brady... Q: Where was Tom Brady born? A: ...

- Comparing probabilities of sequences (e.g. Winograd Schema Challenge):

The cat couldn't fit into the hat because it was too big.  
Does it = the cat or the hat?

≡ Is  $P(\dots \text{because } \mathbf{\text{the cat}} \text{ was too big}) \geq P(\dots \text{because } \mathbf{\text{the hat}} \text{ was too big})$ ?

[Radford et al., 2019]

6      **GPT-2 beats SoTA on language modeling benchmarks with no task-specific fine-tuning**

# Emergent abilities of large language models: GPT-3 (2020)

Language Models are Few-Shot Learners [Brown et al., 2020](#)

GPT-3 (175B parameters; [Brown et al., 2020](#))

- Another increase in size (1.5B -> 175B)
  - And data (40GB -> over 600GB)
- 
- Specify a task by simply **prepend**ing examples of the task before your example
  - Also called **in-context learning**, to stress that *no gradient updates* are performed when learning a new task (there is a separate literature on few-shot learning with gradient updates)

# Limits of prompting for harder tasks?

Some tasks seem too hard for even large LMs to learn through prompting alone.

Especially tasks involving **richer, multi-step reasoning**.

(Humans struggle at these tasks too!)

$$19583 + 29534 = 49117$$

$$98394 + 49384 = 147778$$

$$29382 + 12347 = 41729$$

$$93847 + 39299 = ?$$

We needed a different  
prompting style.

# Chain-of-Thought

## Standard prompting

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: John takes care of 10 dogs. Each dog takes .5 hours a day to walk and take care of their business. How many hours a week does he spend taking care of dogs?

A:

Model output: The answer is 50. ❌

## Chain of thought prompting

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: John takes care of 10 dogs. Each dog takes .5 hours a day to walk and take care of their business. How many hours a week does he spend taking care of dogs?

A:

Model output: John takes care of 10 dogs. Each dog takes .5 hours a day to walk and take care of their business. So that is  $10 \times .5 = 5$  hours a day. 5 hours a day  $\times$  7 days a week = 35 hours a week.  
The answer is 35 hours a week. ✓

# Zero-shot CoT also works!

## 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. ✓

Q: A juggler can juggle 16 balls. Half of the balls are golf balls, and half of the golf balls are blue. How many blue golf balls are there?

A: **Let's think step by step.** There are 16 balls in total. Half of the balls are golf balls. That means there are 8 golf balls. Half of the golf balls are blue. That means there are 4 blue golf balls. ✓

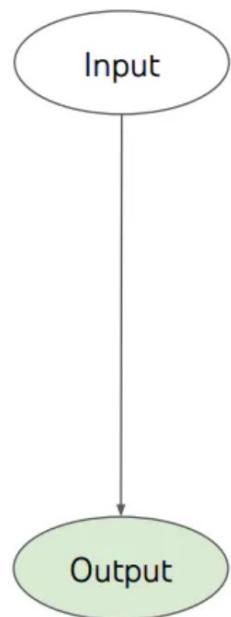
[Kojima et al., 2022]

# Zero-shot chain-of-thought prompting

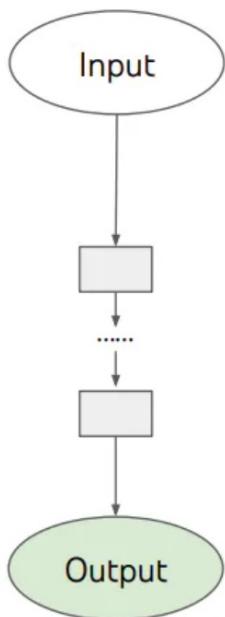
		MultiArith	GSM8K
<b>Zero-Shot</b>		<b>17.7</b>	<b>10.4</b>
Few-Shot (2 samples)		33.7	15.6
Few-Shot (8 samples)		33.8	15.6
<b>Zero-Shot-CoT</b>	<b>Greatly outperforms → 78.7</b>	<b>40.7</b>	
Few-Shot-CoT (2 samples)	<b>zero-shot</b>	84.8	41.3
Few-Shot-CoT (4 samples : First) (*1)		89.2	-
Few-Shot-CoT (4 samples : Second) (*1)	<b>Manual CoT → 90.5</b>	-	-
Few-Shot-CoT (8 samples)	<b>still better → 93.0</b>	48.7	

[[Kojima et al., 2022](#)]

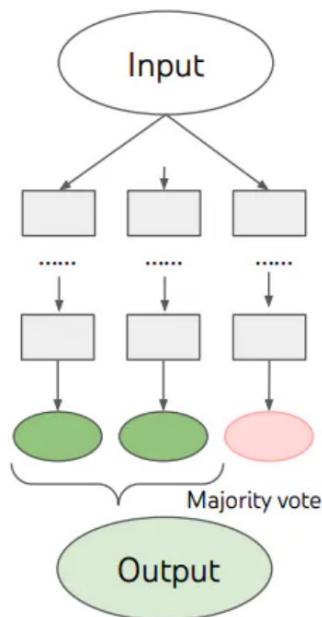
# The new dark art of “prompt engineering”?



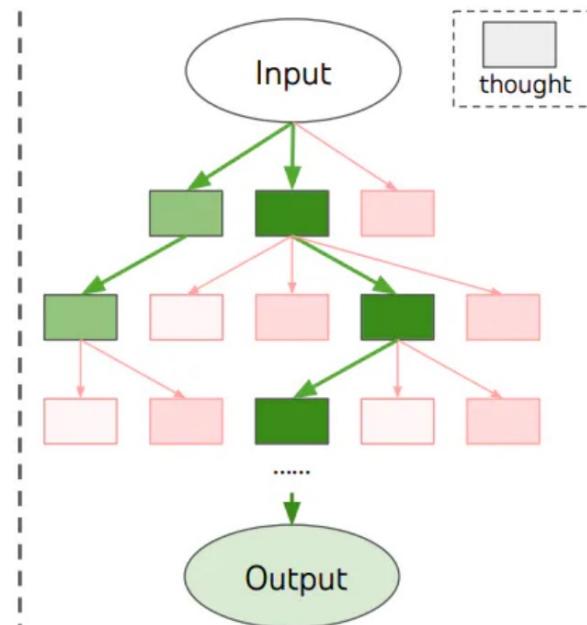
(a) Input-Output  
Prompting (IO)



(c) Chain of Thought  
Prompting (CoT)



(c) Self Consistency  
with CoT (CoT-SC)

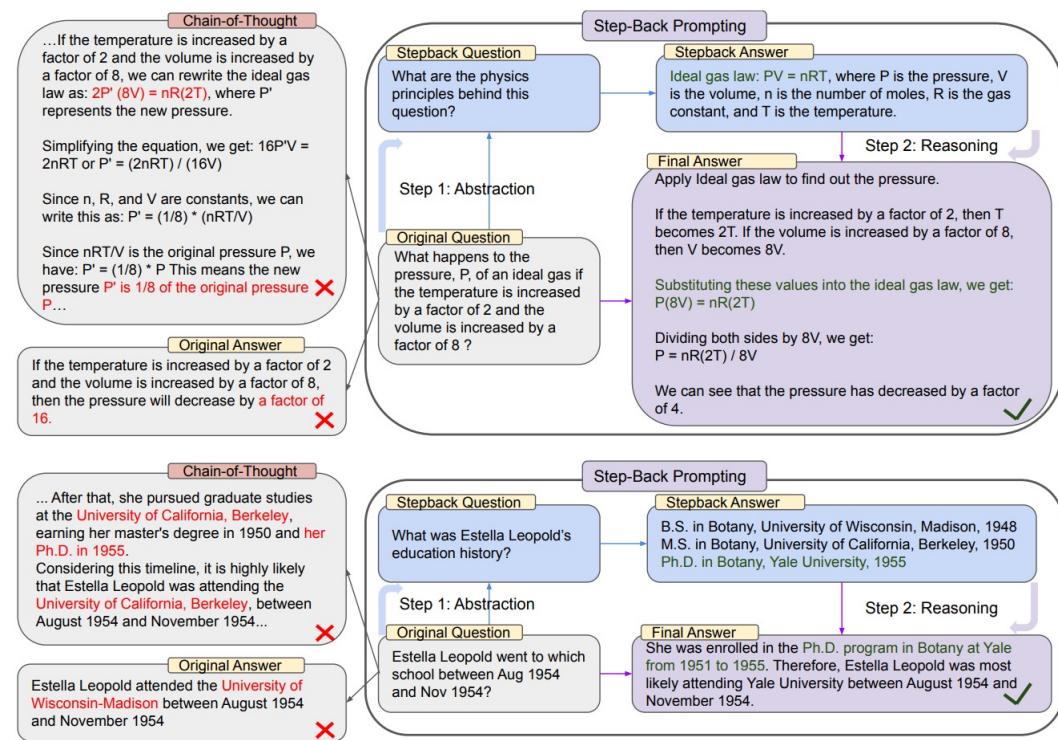
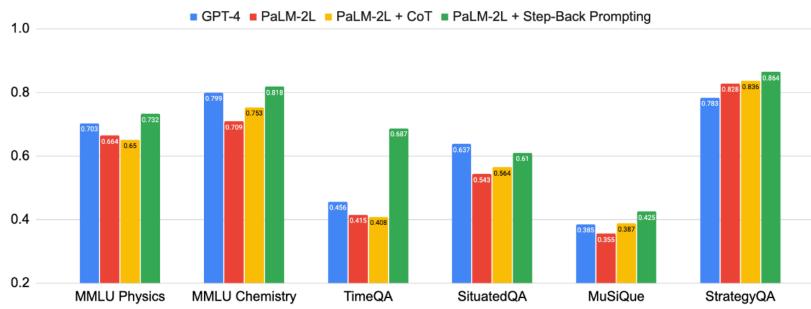


(d) Tree of Thoughts (ToT)

<https://arxiv.org/pdf/2305.10601.pdf>

There are several ways of prompting LLMs and it is a grown space. See this for a good list and overview:  
<https://www.promptingguide.ai/techniques>

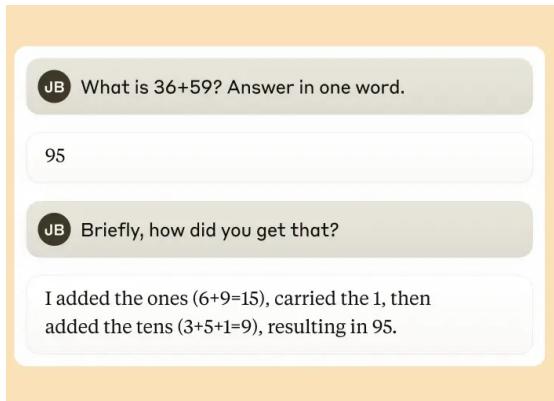
# Step-back Prompting



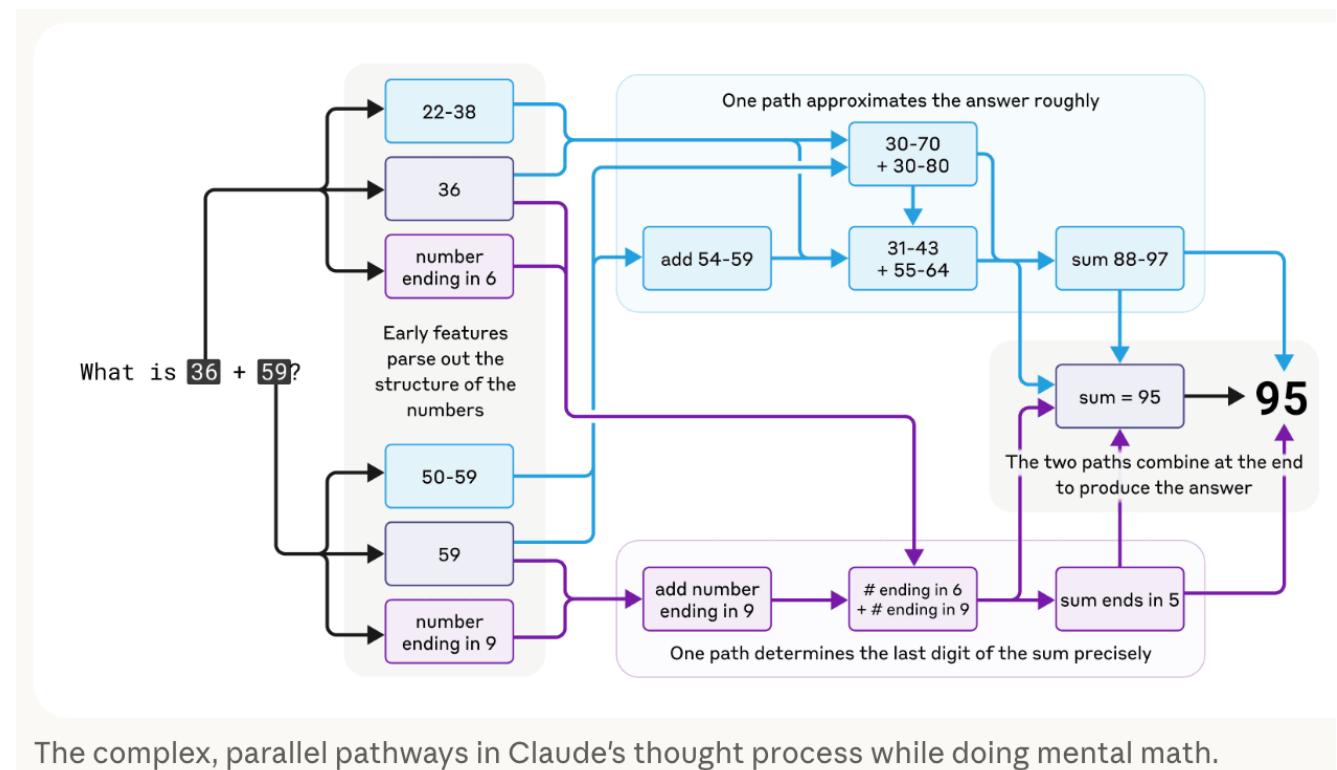
[Take a Step Back: Evoking Reasoning via Abstraction in Large Language Models \(Zheng, et al. 2024\)](#)

# Is CoT truly reflecting how the LLM “thinks”?

What is  $36+59$ ?



What the model says it is doing



# Retrieval Augmented Prompting

Sometimes it helps to provide an explicit context, related to the question.

Sometimes the knowledge is sitting outside the training data/domain of LLMs.

Sometimes we need explicit grounding prior to generating an answer.

So we first retrieve what is potentially relevant/needed and then use them as context in the prompt.

This is called retrieval augmentation.

We will cover this in the next lecture.

# Pros/Cons of in-context learning

- No finetuning needed, prompt engineering (e.g. CoT) can improve performance
- Limits to what you can fit in context and how good the model utilises its context
- LLMs exhibit biases and sensitivity to the selection/order of few-shot demonstrations and wordings of prompts!

Calibrate Before Use: Improving Few-Shot Performance of Language Models: <https://arxiv.org/pdf/2102.09690>

Rethinking Calibration for In-Context Learning and Prompt Engineering: <https://openreview.net/pdf?id=L3FHM0KZcS>

Fantastically Ordered Prompts and Where to Find Them: <https://aclanthology.org/2022.acl-long.556.pdf>

# Language modeling ≠ following user's instruction

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.

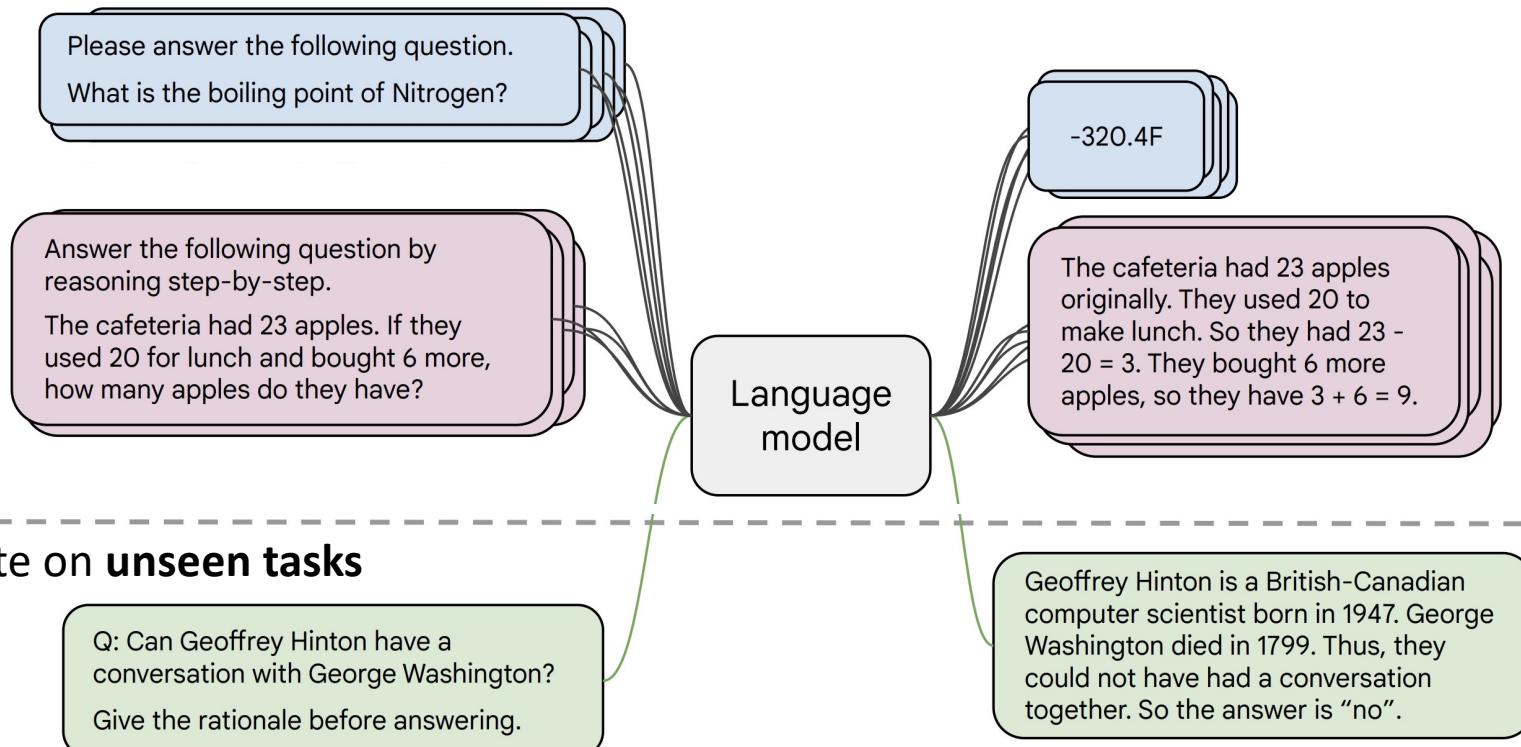
<https://openai.com/research/instruction-following>

Pre-training LMs does not *align them* with user intent [[Ouyang et al., 2022](#)].

Finetuning to the rescue!

# Instruction finetuning

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



[FLAN-T5; [Chung et al., 2022](#)]

# Instruction finetuning

## 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.

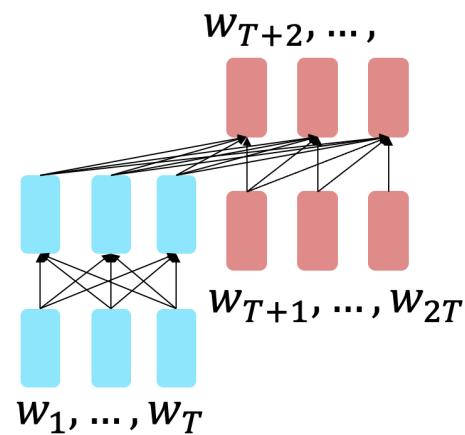
✗ (doesn't answer question)

## 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). ✓

# Instruction finetuning

- Recall the T5 encoder-decoder model from lecture 9 [Raffel et al., 2018], pretrained on the **span corruption** task
- **Flan-T5** [Chung et al., 2022]: T5 models finetuned on 1.8K additional tasks



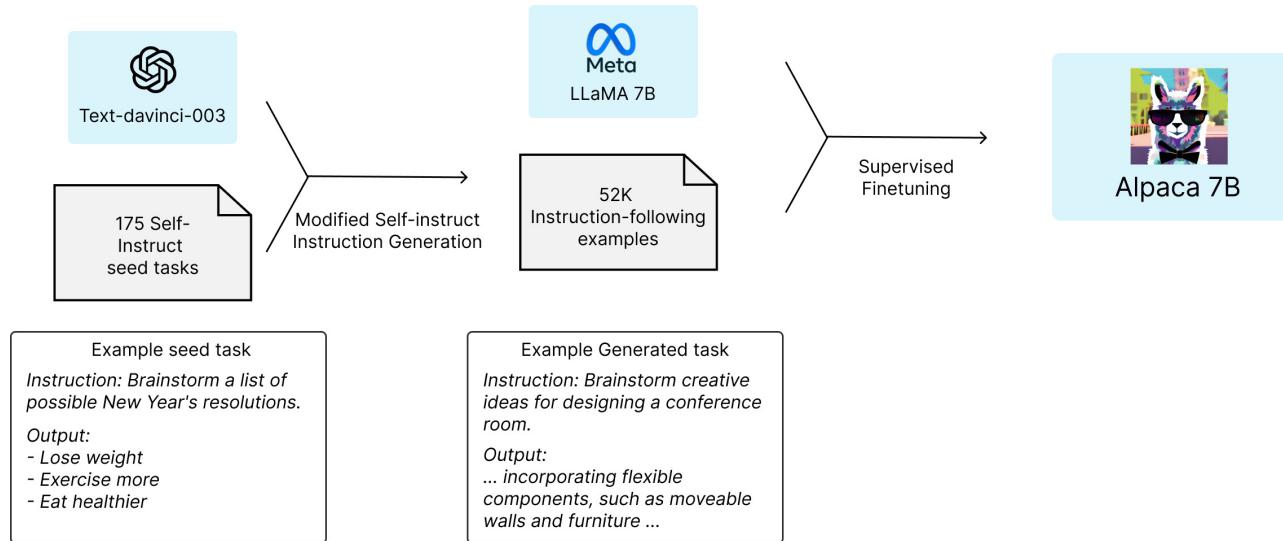
Params	Model	Direct
80M	T5-Small	26.7
	Flan-T5-Small	28.7
250M	T5-Base	25.7
	Flan-T5-Base	35.9
780M	T5-Large	25.1
	Flan-T5-Large	45.1
3B	T5-XL	25.7
	Flan-T5-XL	52.4
11B	T5-XXL	25.9
	Flan-T5-XXL	55.1

Bigger model results in  
bigger improvements

[Chung et al., 2022]

<https://huggingface.co/google/flan-t5-xxl>

# Alpaca Models (Small Scale LMs that follow instructions)



- It turned out that we could actually teach much smaller language models (i.e., 7B parameter) to follow our instructions and exhibit similar behaviour to LLMs (i.e., 175B).
- Alpaca is a language model fine-tuned using supervised learning from a LLaMA 7B model on 52K instruction-following demonstrations generated from OpenAI's text-davinci-003.

# How much instruction tuning data was needed?

LIMA, a 65B parameter LLaMa language model (not fine-tuned with the standard supervised loss on only 1,000 **carefully** curated prompts and responses.

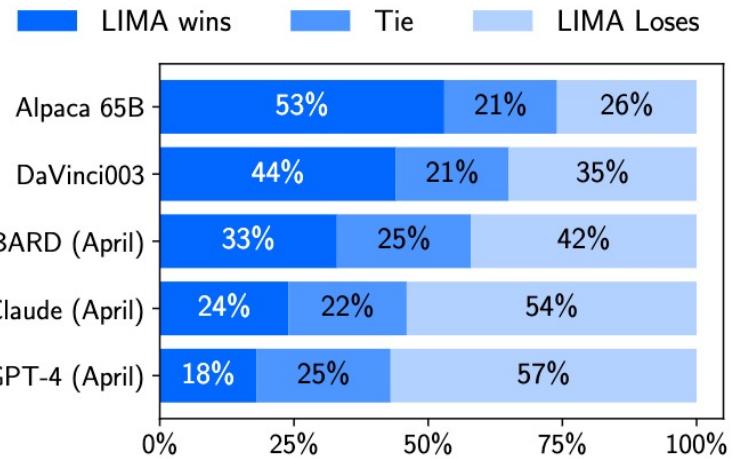


Figure 1: Human preference evaluation, comparing LIMA to 5 different baselines across 300 test prompts.

*“... these results strongly suggest that **almost all knowledge** in large language models is learned during pretraining, and only limited instruction tuning data is necessary to teach models to produce high quality output.”*

# Pros/Cons of instruction tuning

- Simple and straightforward and close to human-like communications
- Seems to also generalize to unseen tasks
- It's expensive to collect ground-truth data for tasks.
- For tasks like open-ended text generation we have no right answer.
- Even with instruction finetuning, there is a mismatch between the LM objective and the objective of "satisfy human preferences"!

Can we **explicitly attempt to satisfy human preferences?**

Also see this paper (for a method to self-generate instruction-tuning data):

SELF-INSTRUCT: Aligning Language Models with Self-Generated Instructions <https://arxiv.org/pdf/2212.10560>

# Why not just engineer a metric for preference?

If your preference could be reliably captured with a set of engineered features or explicit metrics, then we probably wouldn't need RL (or RLHF) at all.

But consider:

- How would you formally specify what makes a *good story*?
- How would you write a metric for a *good summary*?
- How would you quantify the quality of an *explanation*?
- How would you quantify helpfulness?
- How would you quantify appropriateness?

# Why not just engineer a metric for preference?

If your preference could be reliably captured with a set of engineered features or explicit metrics, then we probably wouldn't need RL (or RLHF) at all.

But consider:

- How would you formally specify what makes a *good story*?
- How would you write a metric for a *good summary*?
- How would you quantify the quality of an *explanation*?
- How would you quantify helpfulness?
- How would you quantify appropriateness?

These preferences are inherently subjective, contextual, and hard to reduce to simple rules. It's unlikely you could sit down and design a fixed metric that generalizes well. Even if you do, it is likely that other people will not agree with your metric.

# Why not just engineer a metric for preference?

If your preference could be reliably captured with a set of engineered features or explicit metrics, then we probably wouldn't need RL (or RLHF) at all.

But consider:

- How would you formally specify what makes a *good story*?
- How would you write a metric for a *good summary*?
- How would you quantify the quality of an *explanation*?
- How would you quantify helpfulness?
- How would you quantify appropriateness?

These preferences are inherently subjective, contextual, and hard to reduce to simple rules. It's unlikely you could sit down and design a fixed metric that generalizes well. Even if you do, it is likely that other people will not agree with your metric.

RLHF is useful when the desired behavior is difficult to encode as an explicit reward function, but humans can reliably judge it. Instead of hand-engineering metrics, we learn a reward model from human preferences.

# Optimizing for human preferences

- Let's say we were training a language model on some task (e.g. summarization).
- For each LM generated sample  $s$ , imagine we had a way to obtain a *human reward* of that summary:  $R(s) \in \mathbb{R}$ , higher is better.

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

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

$$s_1 \\ R(s_1) = 8.0$$

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

$$s_2 \\ R(s_2) = 1.2$$

# Optimizing for human preferences

- Let's say we were training a language model on some task (e.g. summarization).
- For each LM generated sample  $s$ , imagine we had a way to obtain a *human reward* of that summary:  $R(s) \in \mathbb{R}$ , higher is better.

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

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

$$\begin{array}{ll} s_1 & s_2 \\ R(s_1) = 8.0 & R(s_2) = 1.2 \end{array}$$

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

- Now we want to optimize the LM such that we maximize the expected reward of outputs from our LM:

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

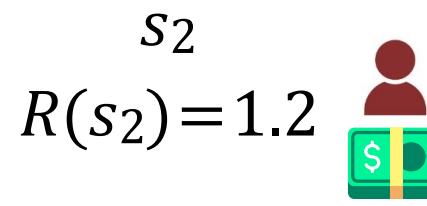
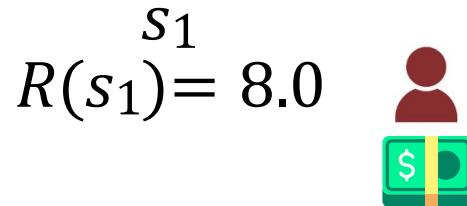
# How do we model human preferences?

- Awesome: now for any **arbitrary, non-differentiable reward function**  $R(s)$ , we can train our language model to maximize expected reward.
- **Problem 1:** human-in-the-loop is expensive!

An earthquake hit  
San Francisco.

There was minor  
property damage,  
but no injuries.

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



# How do we model human preferences?

- Awesome: now for any **arbitrary, non-differentiable reward function**  $R(s)$ , we can train our language model to maximize expected reward.
- **Problem 1:** human-in-the-loop is expensive!
  - **Solution:** instead of directly asking humans for preferences, **model their preferences** as a separate (NLP) problem!

An earthquake hit  
San Francisco.

There was minor  
property damage,  
but no injuries.

$$S_1 \\ R(S_1) = 8.0$$


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

$$S_2 \\ R(S_2) = 1.2$$


**Basic Regression Problem:**  
Train an LM  $RM_\phi(s)$  to  
predict human  
preferences from an  
annotated dataset, then  
optimize for  $RM_\phi$  instead.

# How do we model human preferences?

- **Problem 2:** human judgments are noisy and miscalibrated!
- **Solution:** instead of asking for direct ratings, ask for **pairwise comparisons**, which can be more reliable

A 4.2 magnitude  
earthquake hit  
San Francisco,  
resulting in  
massive damage.

$s_3$

$$R(s_3) = \begin{matrix} 4.1? & 6.6? & 3.2? \end{matrix}$$

# How do we model human preferences?

- **Problem 2:** human judgments are noisy and miscalibrated!
- **Solution:** instead of asking for direct ratings, ask for **pairwise comparisons**, which can be more reliable

An earthquake hit  
San Francisco.

There was minor  
property damage,  
but no injuries.

$s_1$

A 4.2 magnitude  
earthquake hit  
San Francisco,  
resulting in  
massive damage.

$s_3$

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

>

$s_2$

# How do we model human preferences?

- **Problem 2:** human judgments are noisy and miscalibrated!
- **Solution:** instead of asking for direct ratings, ask for **pairwise comparisons**, which can be more reliable

An earthquake hit  
San Francisco.

There was minor  
property damage,  
but no injuries.

$s_1$

A 4.2 magnitude  
earthquake hit

San Francisco,  
resulting in  
massive damage.

$s_3$

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

>

>

$s_2$

Bradley-Terry [1952] paired comparison model

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

“winning”  
sample

“losing”  
sample

$s^w$  should score  
higher than  $s^l$

# RLHF: Putting it all together

- Finally, we have everything we need:
  - A pretrained (possibly instruction-finetuned) LM  $p^{PT}(s)$
  - A reward model  $RM_{\phi}(s)$  that produces scalar rewards for LM outputs, trained on a dataset of human comparisons
  - A method for optimizing LM parameters towards an arbitrary reward function.
- Now to do RLHF:
  - Initialize a copy of the model  $p_{\theta}^{RL}(s)$ , with parameters  $\theta$  we would like to optimize

$$\mathbb{E}_{\hat{s} \sim P_{\theta}^{RL}(s)} [RM_{\phi}(\hat{s})]$$

Do you foresee any issue with this?

# RLHF: Putting it all together

- Finally, we have everything we need:
  - A pretrained (possibly instruction-finetuned) LM  $p^{PT}(s)$
  - A reward model  $RM_{\phi}(s)$  that produces scalar rewards for LM outputs, trained on a dataset of human comparisons
  - A method for optimizing LM parameters towards an arbitrary reward function.
- Now to do RLHF:
  - Initialize a copy of the model  $p_{\theta}^{RL}(s)$ , with parameters  $\theta$  we would like to optimize

$$\mathbb{E}_{\hat{s} \sim P_{\theta}^{RL}(s)} [RM_{\phi}(\hat{s})]$$

Do you foresee any issue with this?

Learned Rewards could be erroneous.

We could overwrite everything the model learned.

## RLHF: Putting it all together

Instead of optimizing this:

$$J(\theta) = \mathbb{E}_{\hat{s} \sim P_{\theta}^{RL}(s)} [RM_{\phi}(\hat{s})]$$

We optimize this:

$$J(\theta) = \mathbb{E}_{\hat{s} \sim P_{\theta}^{RL}(s)} [RM_{\phi}(\hat{s})] - \beta \times \underbrace{\mathbb{E}_{\hat{s} \sim P_{\theta}^{RL}(s)} [\log \left( \frac{P_{\theta}^{RL}(\hat{s})}{P^{PT}(\hat{s})} \right)]}$$

Pay a price when  
 $p_{\theta}^{RL}(s) > p^{PT}(s)$

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

# RLHF: Putting it all together

Instead of optimizing this:

$$J(\theta) = \mathbb{E}_{\hat{s} \sim P_{\theta}^{RL}(s)} [RM_{\phi}(\hat{s})]$$

We optimize this:

$$J(\theta) = \mathbb{E}_{\hat{s} \sim P_{\theta}^{RL}(s)} [RM_{\phi}(\hat{s})] - \beta \times \underbrace{\mathbb{E}_{\hat{s} \sim P_{\theta}^{RL}(s)} [\log \left( \frac{P_{\theta}^{RL}(\hat{s})}{P^{PT}(\hat{s})} \right)]}$$

Pay a price when  
 $p_{\theta}^{RL}(s) > p^{PT}(s)$

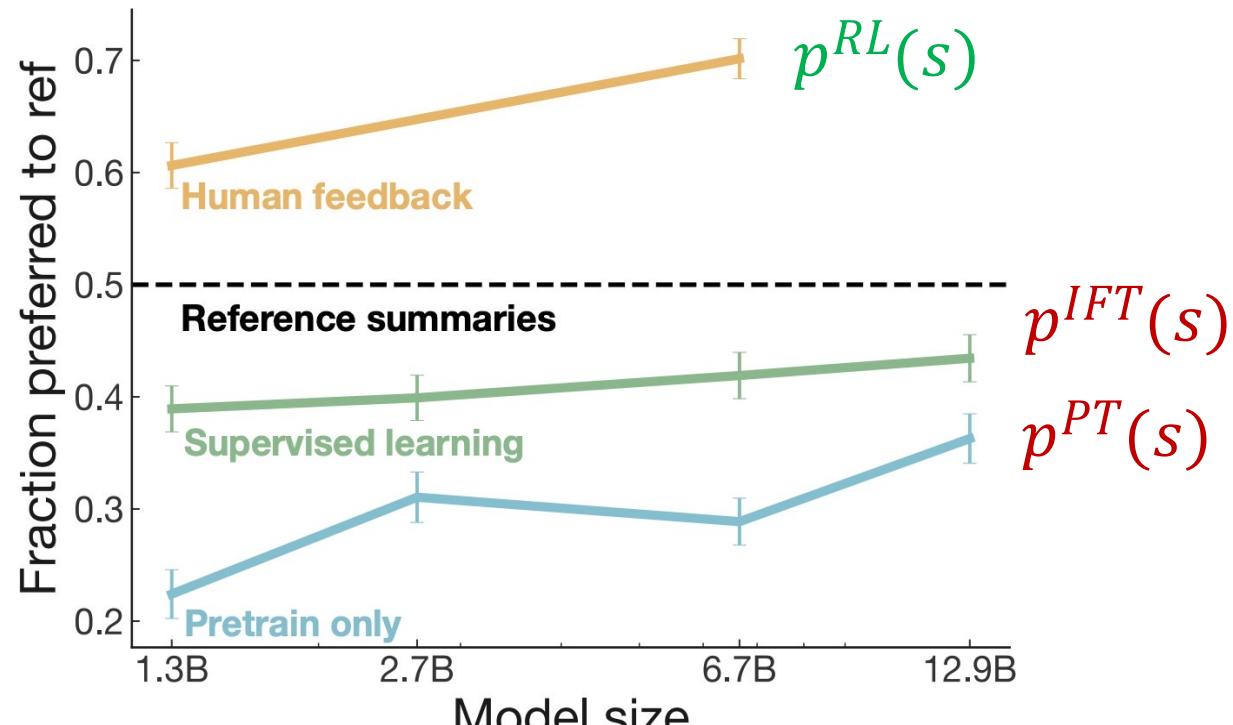
This is a penalty which prevents us from diverging too far from the pretrained model. In expectation, it is known as the **Kullback-Leibler (KL)** divergence between  $p_{\theta}^{RL}(s)$  and  $p^{PT}(s)$

How do we optimize this? We use Reinforcement Learning (i.e., the PPO Algorithm).

$$\theta \leftarrow \theta + \eta \nabla_{\theta} J(\theta)$$

$$\nabla_{\theta} J(\theta) = \nabla_{\theta} \mathbb{E}_{s \sim P_{\theta}^{RL}} [RM_{\phi}(s)] - \beta \nabla_{\theta} \mathbb{E}_{s \sim P_{\theta}^{RL}} \left[ \log \frac{P_{\theta}^{RL}(s)}{P^{PT}(s)} \right]$$

# RLHF provides gains over pretraining + finetuning



[Stiennon et al., 2020]

# InstructGPT: scaling up RLHF to tens of thousands of tasks

**30k  
tasks!**

Step 1

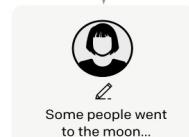
**Collect demonstration data,  
and train a supervised policy.**

A prompt is sampled from our prompt dataset.

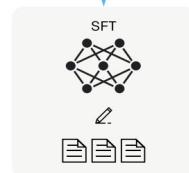


Some people went to the moon...

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) Explain gravity...  
(B) Explain war...

(C) Moon is natural satellite of...

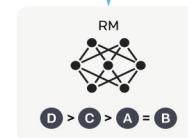
(D) People went to the moon...



D > C > A = B

A labeler ranks the outputs from best to worst.

This data is used to train our reward model.



D > C > A = B

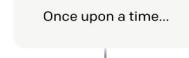
Step 3

**Optimize a policy against the reward model using reinforcement learning.**

A new prompt is sampled from the dataset.



The policy generates an output.



r<sub>k</sub>

The reward model calculates a reward for the output.

The reward is used to update the policy using PPO.

<https://openai.com/index/instruction-following>

# InstructGPT: scaling up RLHF to tens of thousands of tasks

Tasks collected from labelers:

- **Plain:** We simply ask the labelers to come up with an arbitrary task, while ensuring the tasks had sufficient diversity.
- **Few-shot:** We ask the labelers to come up with an instruction, and multiple query/response pairs for that instruction.
- **User-based:** We had a number of use-cases stated in waitlist applications to the OpenAI API. We asked labelers to come up with prompts corresponding to these use cases.

Use-case	Prompt
Brainstorming	List five ideas for how to regain enthusiasm for my career
Generation	Write a short story where a bear goes to the beach, makes friends with a seal, and then returns home.

[Ouyang et al., 2022]

# InstructGPT

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.

# Can we simplify RLHF? Direct Preference Optimization

What if there was a way to write  $RM_\phi(x, y)$  in terms of  $p_\theta^{RL}(\hat{y} \mid x)$ ?

....

See this for full derivations: <https://arxiv.org/pdf/2305.18290>

$$RM_\theta(x, \hat{y}) = \beta \log \frac{p_\theta^{RL}(\hat{y} \mid x)}{p^{PT}(\hat{y} \mid x)} + \beta \log Z(x)$$

# RLHF without explicit Reward Modelling = DPO

- Recall, how we fit the reward model  $RM_\phi(x, y)$  :

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

- Notice that we only need the **difference** between the rewards for  $\mathbf{y}^w$  and  $\mathbf{y}^l$ . Simplify for  $RM_\theta(x, y)$ :

$$RM_\theta(x, \mathbf{y}^w) - RM_\theta(x, \mathbf{y}^l) = \beta \log \frac{p_\theta^{RL}(\mathbf{y}^w | x)}{p_\theta^{PT}(\mathbf{y}^w | x)} - \beta \log \frac{p_\theta^{RL}(\mathbf{y}^l | x)}{p_\theta^{PT}(\mathbf{y}^l | x)}$$

- The final DPO loss function is:

$$J_{DPO}(\theta) = -\mathbb{E}_{(x, \mathbf{y}^w, \mathbf{y}^l) \sim D} [\log \sigma(RM_\theta(x, \mathbf{y}^w) - RM_\theta(x, \mathbf{y}^l))]$$

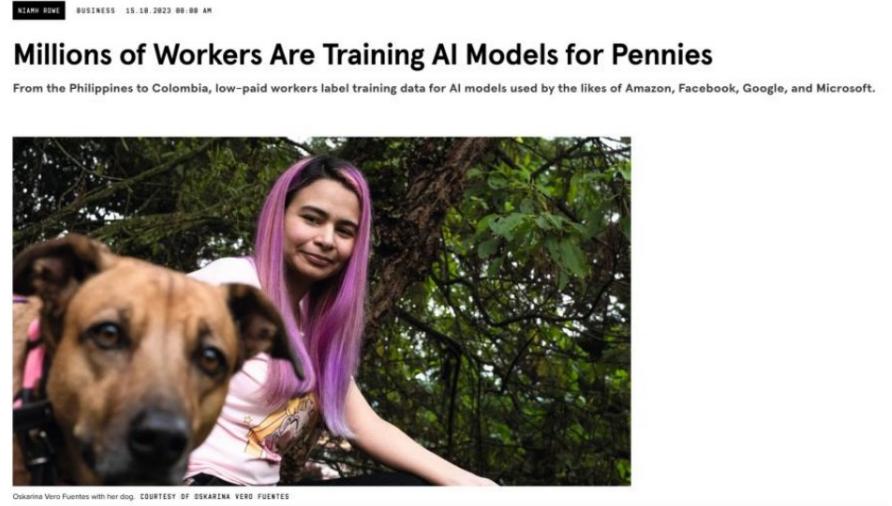
We have a *simple classification loss* function that connects **preference data to language model parameters** directly!

[https://huggingface.co/docs/trl/main/en/dpo\\_trainer](https://huggingface.co/docs/trl/main/en/dpo_trainer)

43

We do not need to use RL any more. Because optimizing the reward model essentially optimizes the language model now.

# Source of labels for preference data



Behind the AI boom, an army of overseas workers in 'digital sweatshops'



# Even-Handedness

## Base Pre-Training Data Issues:

- English-language dominance reflects Western cultural values
- Underrepresentation of non-Western perspectives

## RLHF/Alignment Stage Amplification:

- If annotators or prompts are not representative, the reward model will encode cultural biases
- Human feedback can inadvertently reinforce demographic or political biases present in the labeler pool

## Some Mitigation Strategies:

- **Constitutional AI:** Using explicit principles rather than relying solely on crowdworker preferences
- **Character training:** Directly training models on specific neutrality traits during alignment
- **System prompts:** Providing explicit instructions for even-handed behavior
- **Diverse evaluation:** Testing across thousands of prompts covering hundreds of political stances

# Pluralistic Alignment

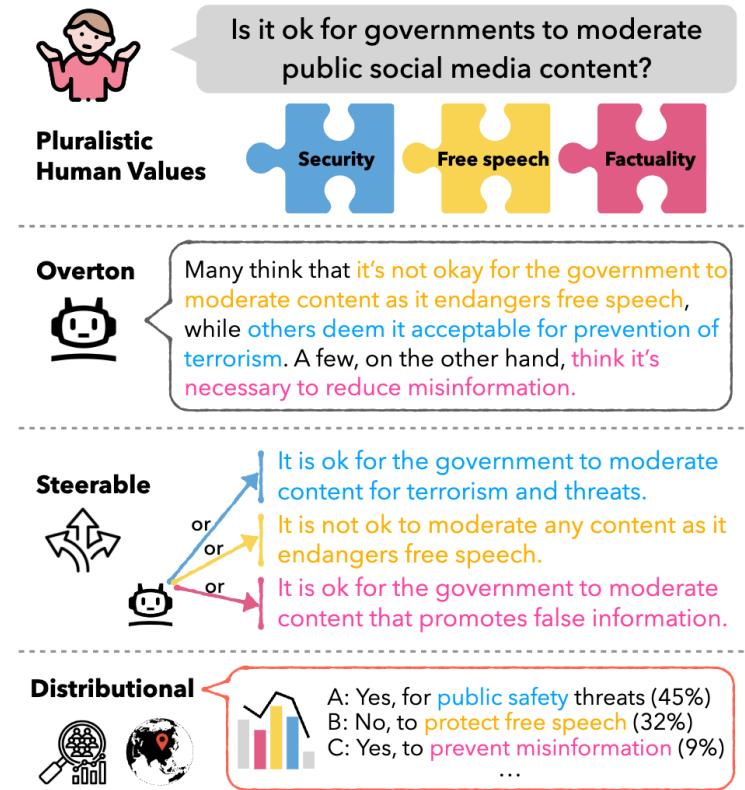


Figure 1. Three kinds of pluralism in models.

# Alignment Tax

A limitation of RLHF is that it introduces an “alignment tax”: aligning the models can make their performance worse on some other tasks. Their solution in InstructGPT was to minimize this alignment tax during RL by mixing in a small fraction of the original data used to train GPT-3, and train on this data using the normal log likelihood maximization.

<https://openai.com/index/instruction-following/>

Methods	Reasoning Accuracy				Harmful Score
	AIME24	GPQA	MATH500	Average	BeaverTails
Base model (Qwen-32B-instruct)	16.67	40.40	65.20	40.76	16.70
LRM (S1.1-32B)	<b>40.00</b>	<b>58.59</b>	<b>91.60</b>	<b>63.40</b>	60.40
LRM + DirectRefusal	13.33	35.35	48.80	32.49	<b>0.80</b>

Safety gains aren’t “free”: enhancing LRM safety trained sequentially imposes a reasoning accuracy cost.

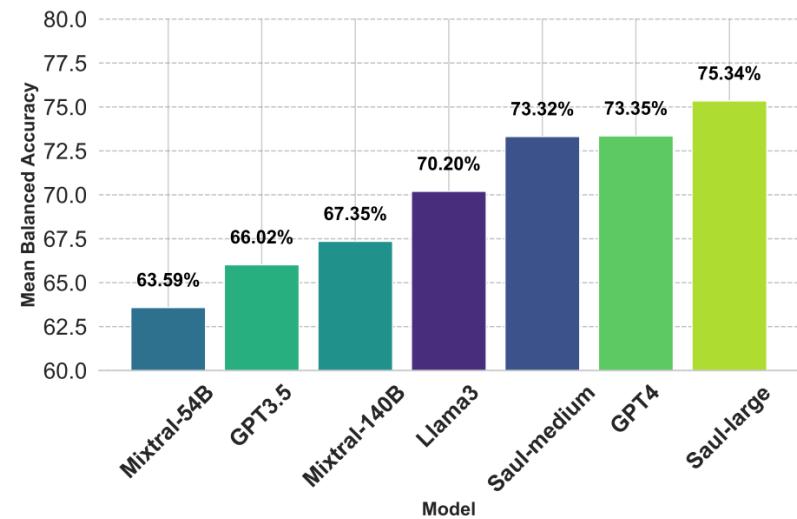
<https://arxiv.org/pdf/2503.00555>

# Steps to build an LLM for a new domain

# LLM for Law

## SaulLM-54B & SaulLM-141B: Scaling Up Domain Adaptation for the Legal Domain

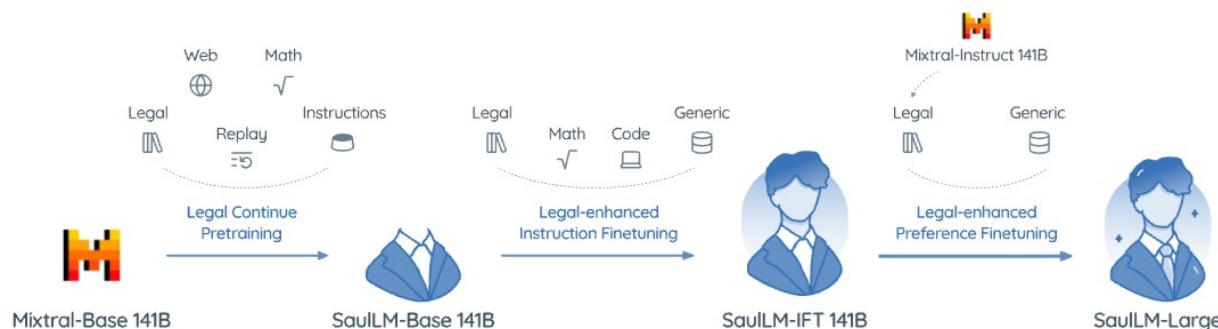
Pierre Colombo Equall MICS - CentraleSupelec	Telmo Pires Equall	Malik Boudiaf Equall	Rui Melo Equall
Dominic Culver Equall	Sofia Morgado Equall	Etienne Malaboeuf CINES	Gabriel Hautreux CINES
Johanne Charpentier CINES		Michael Desa Equall	



# LLM for Law

3 Key steps as you expect:

- Pretraining
- Instruction tuning
- Preference tuning



The computational backbone for the continuous pretraining phase of our project consists of 384 AMD MI250 GPUs.

Table 1: Sources of Legal Pretraining Data

Source Name	Tokens (B)
FreeLaw Subset from The Pile	15
EDGAR Database	5
English MultiLegal Pile	50
English EuroParl	6
GovInfo Statutes, Opinions & Codes	11
Law Stack Exchange	0.019
Comm Open Australian Legal Corpus	0.5
EU Legislation	0.315
UK Legislation	0.190
Court Transcripts	0.350
UPSTO Database	4.7
Web Data (legal)	400
Other	30
<b>Total</b>	<b>520</b>

### Pretraining

**Scale & coverage** – An English-language corpus of ~500 billion raw tokens is gathered from multiple jurisdictions (US, EU, Australia, UK, etc.) to capture common-law and civil-law traditions.

**Replay & maths add-ons** – To avoid catastrophic forgetting, generic text (Wikipedia, StackExchange, GitHub) and maths was replayed.

### Instruction Tuning

**General instructions** – About **1 M items** drawn from UltraInteract and Dolphin.

**Legal instructions** – Synthetic multi-turn Q&A dialogues generated with Mistral-54B/141B-Instruct. Each script starts with a user query about a legal document, then unfolds the legal reasoning through follow-up questions.

### Preference Tuning

**General** – Pairwise preference datasets such as UltraFeedback and Orca.

**Legal-specific** – Synthetic legal scenarios; responses are scored for factuality, relevance and logic by Mixtral-142B-Instruct to create accepted/rejected pairs for Direct Preference Optimisation (DPO).

# LLM for Law – You still need to specialise to task

## Example 1.

Query: *The distinction between a genuine offer of compromise and a demand to capitulate has to be recognised. See the discussion in [CITATION].*

Answer: *Leichhardt Municipal Council v Green [2004] NSWCA 341*

## Example 2.

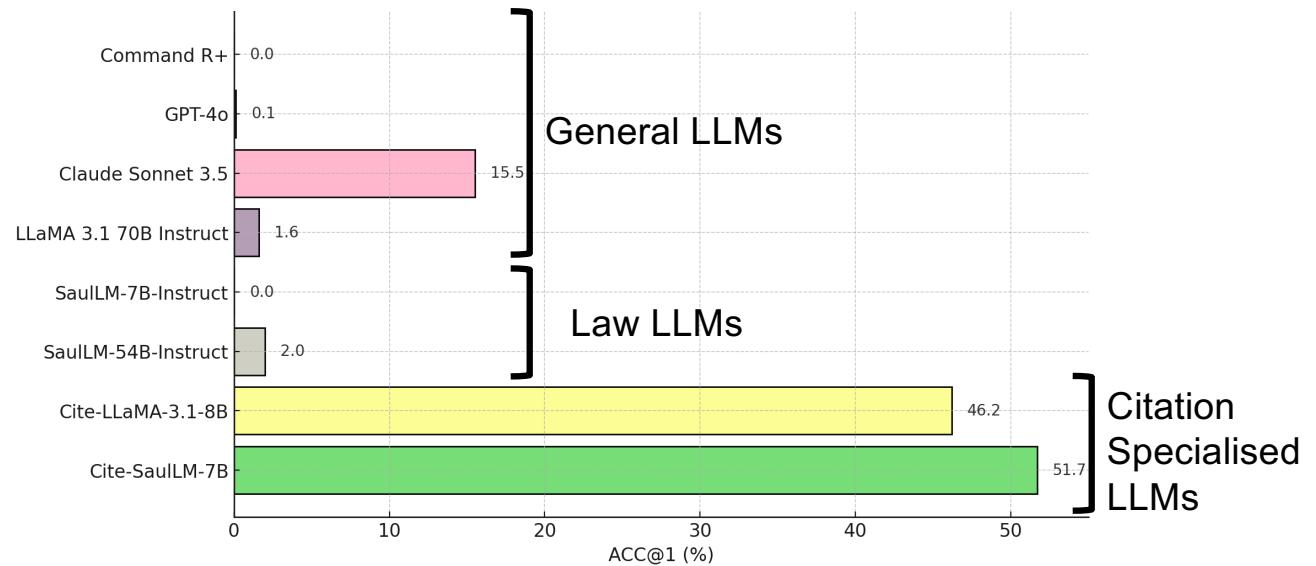
Query: *The Tribunal is satisfied that the applicant does not fulfil the requirements of section 139(a) of the National Law, in that she lacks the mental capacity to practise medicine, as was considered in [CITATION].*

Answer: *Lindsay v Health Care Complaints Commission [2010] NSWCA 194*

## Example 3.

Query: *Whilst it is suggested that the offender's mother and grandmother have difficulty paying rent without the offender's assistance, there is no evidence of how they support themselves or their financial circumstances. There is no evidence of hardship that might meet the 'truly, wholly or highly exceptional' standard referred to in [CITATION].*

Answer: *Jinnette v R [2012] NSWCCA 217*



<https://auslawbench.github.io>