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 Pacheo
- 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[1] **Tim Rocktäschel**[1,2] **Patrick Lewis**[1,2] **Anton Bakhtin**[1]
Yuxiang Wu[1,2] **Alexander H. Miller**[1] **Sebastian Riedel**[1,2]
[1]Facebook AI Research
[2]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[1] FABIOPETRONI@FB.COM
Patrick Lewis[1,2] PLEWIS@FB.COM
Aleksandra Piktus[1] PIKTUS@FB.COM
Tim Rocktäschel[1,2] ROCKT@FB.COM
Yuxiang Wu[2] YUXIANG.WU.18@UCL.AC.UK
Alexander H. Miller[1] AHM@FB.COM
Sebastian Riedel[1,2] SRIEDEL@FB.COM
[1] *Facebook AI Research*
[2] *University College London*

May 2020

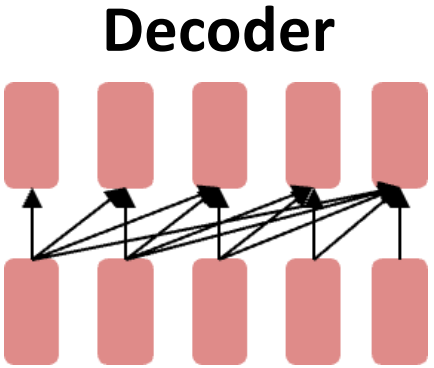
Rewire-then-Probe: A Contrastive Recipe for Probing Biomedical Knowledge of Pre-trained Language Models

Zaiqiao Meng[♠◇*] **Fangyu Liu**[♠*] **Ehsan Shareghi**[♠♣]
Yixuan Su[♠] **Charlotte Collins**[♠] **Nigel Collier**[♠]
[♠]Language Technology Lab, University of Cambridge
[◇]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.

entailment

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

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 [*1] Jeffrey Wu [*1] Rewon Child [1] David Luan [1] Dario Amodei [**1] Ilya Sutskever [**1]

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(...because` **`the cat`** `was too big) >=`
  `P(...because` **`the hat`** `was too big)`?

[Radford et al., 2019]

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 **prepending 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.

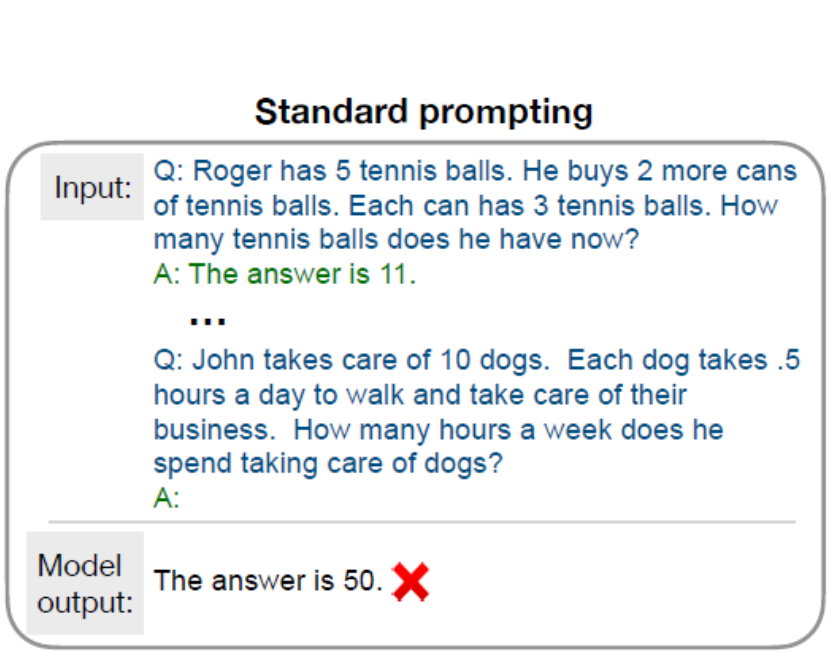
Especially tasks involving **richer, multi-step reasoning.**

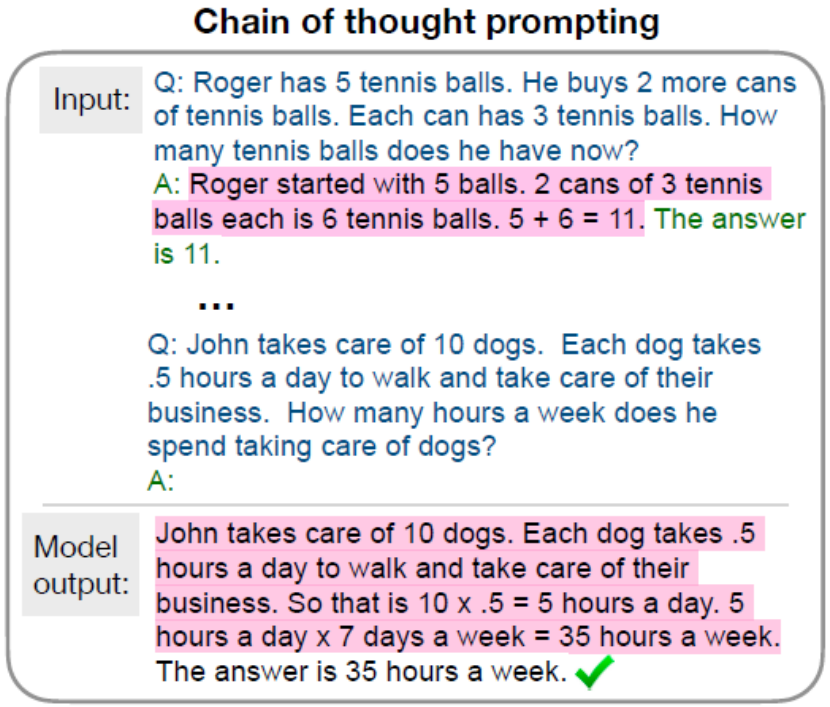
(Humans struggle at these tasks too!)

```
19583 + 29534 = 49117
98394 + 49384 = 147778
29382 + 12347 = 41729
93847 + 39299 = ?
```

We needed a different prompting style.

Chain-of-Thought





https://arxiv.org/abs/2201.11903

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

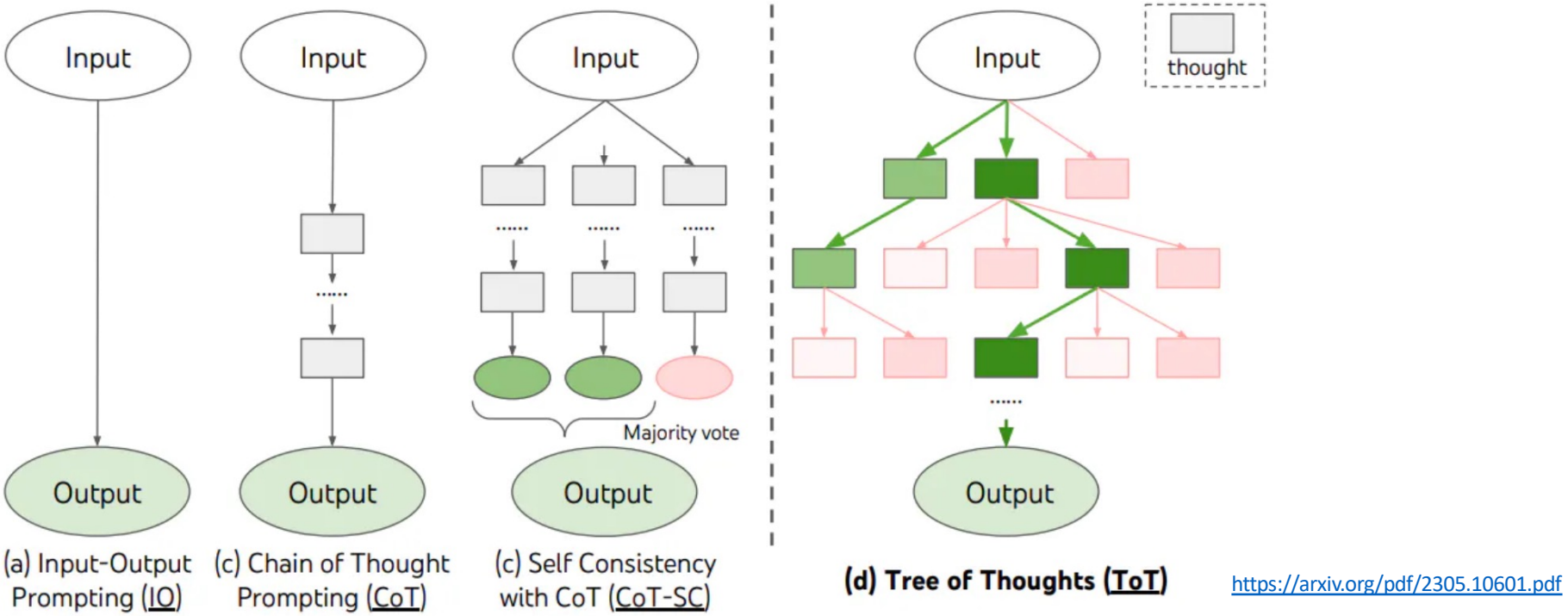
	MultiArith	GSM8K
Zero-Shot	**17.7**	**10.4**
Few-Shot (2 samples)	33.7	15.6
Few-Shot (8 samples)	33.8	15.6
Zero-Shot-CoT	**78.7**	**40.7**
Few-Shot-CoT (2 samples)	84.8	41.3
Few-Shot-CoT (4 samples : First) (*1)	89.2	-
Few-Shot-CoT (4 samples : Second) (*1)	90.5	-
Few-Shot-CoT (8 samples)	93.0	48.7

Greatly outperforms zero-shot →

Manual CoT still better →

[Kojima et al., 2022]

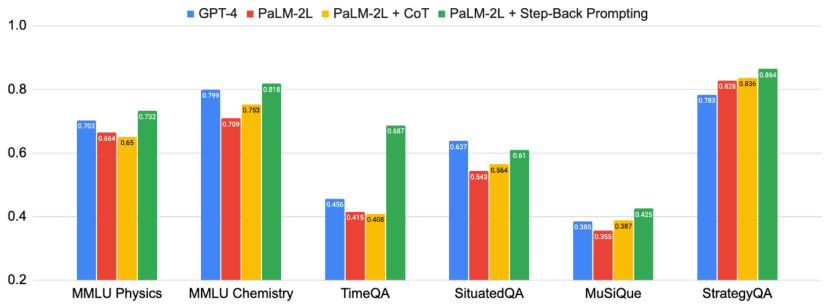
The new dark art of "prompt engineering"?

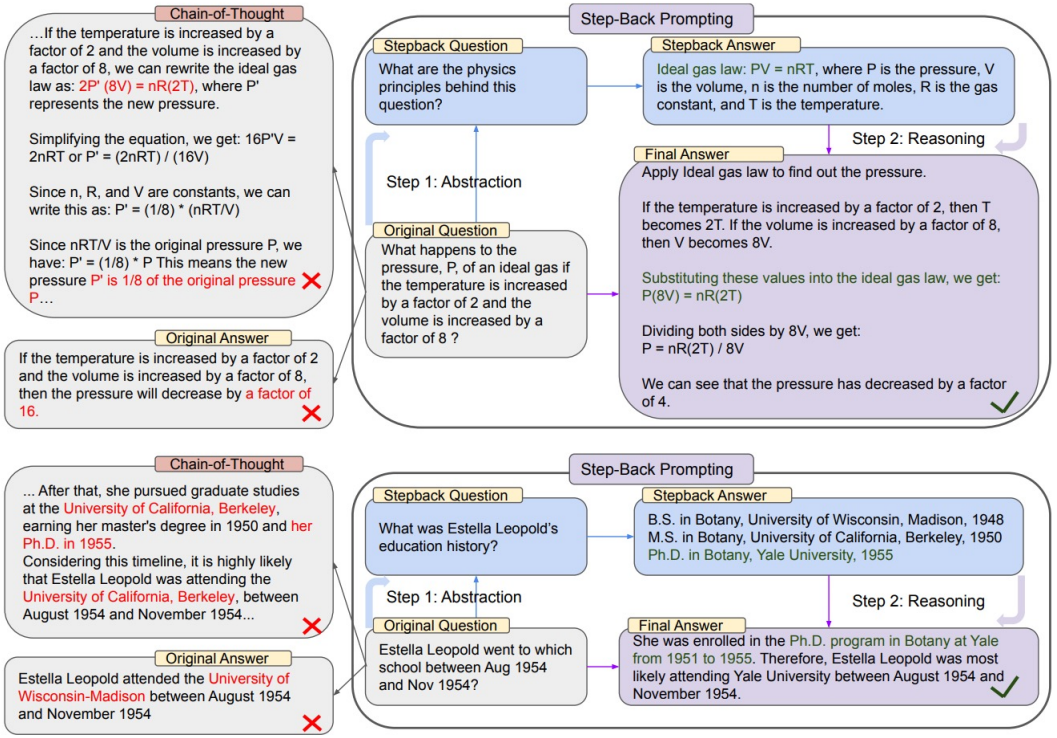


https://arxiv.org/pdf/2305.10601.pdf

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

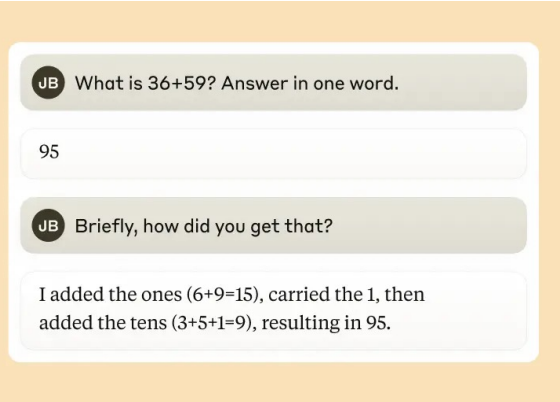




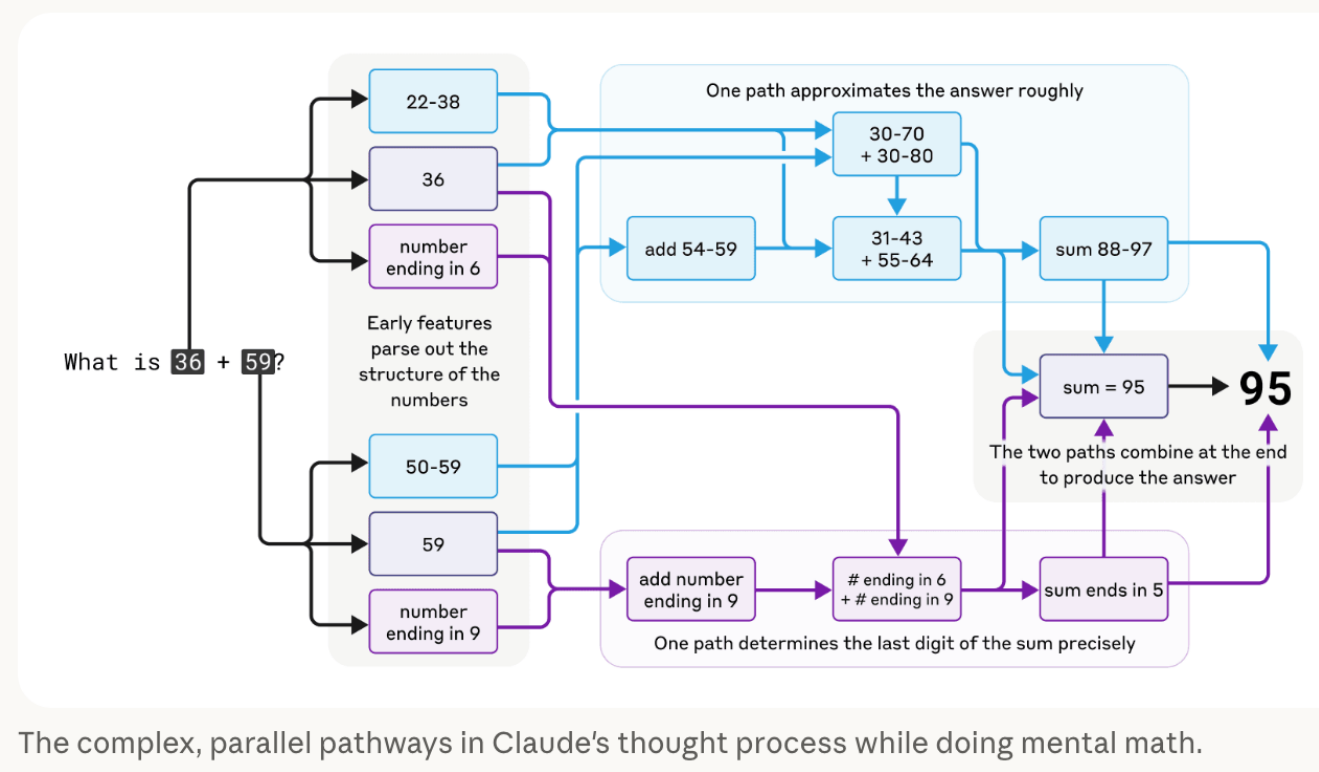
[Take a Step Back: Evoking Reasoning via Abstraction in Large Language Models (Zheng, et al. 2024)]

Is CoT truly reflecting how the LLM "thinks"?

What is 36+59?



What the model says it is doing



The complex, parallel pathways in Claude's thought process while doing mental math.

What the model does under the hood

https://www.anthropic.com/research/tracing-thoughts-language-model

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=L3FHMoKZcS
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.
```

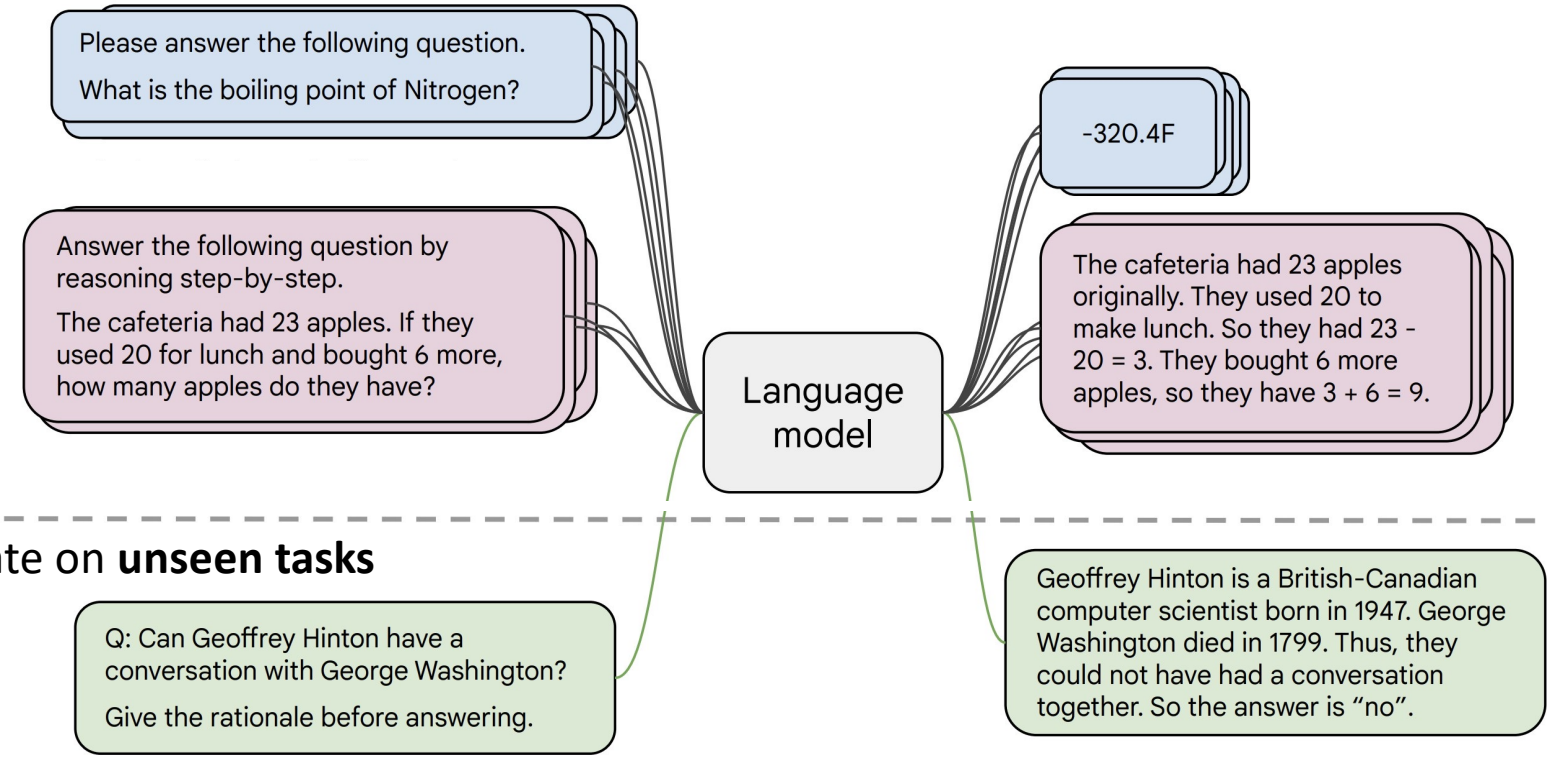
https://openai.com/research/instruction-following

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

Finetuning to the rescue!

Instruction finetuning

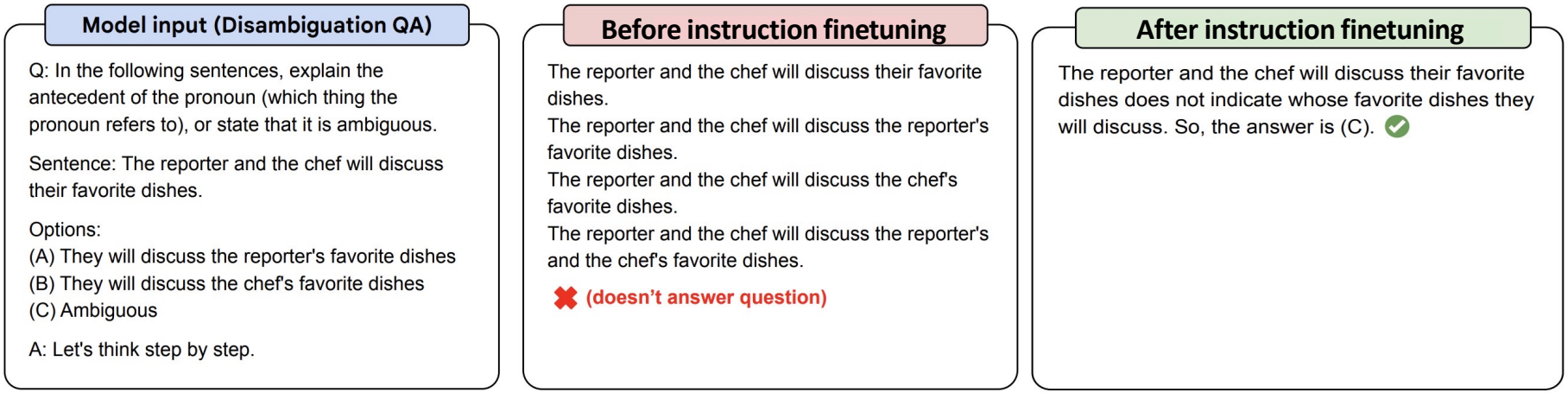
- **Collect examples** of (instruction, output) pairs across many tasks and finetune an LM



- Evaluate on **unseen tasks**

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

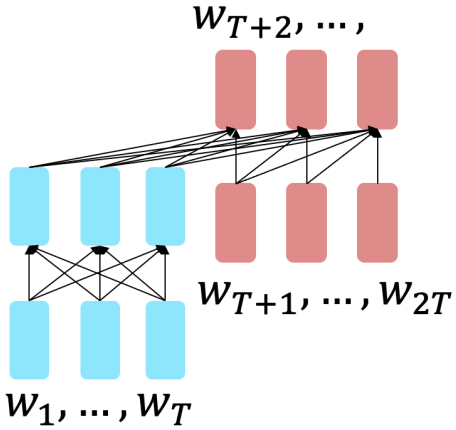
Instruction finetuning



[Chung et al., 2022]

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



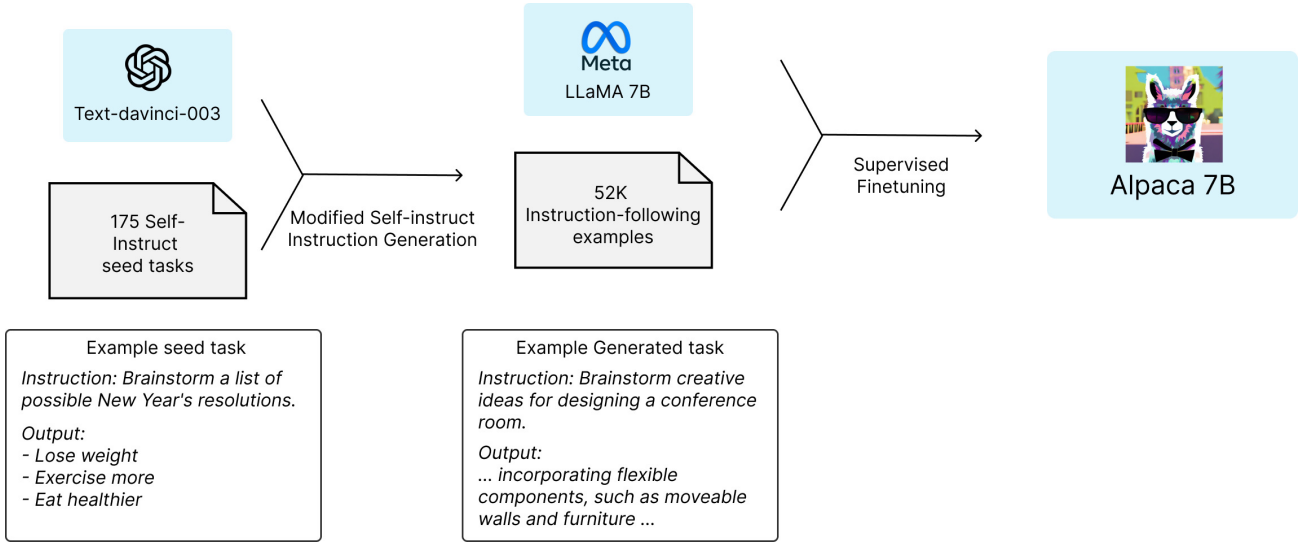
https://huggingface.co/google/flan-t5-xxl

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]

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, where outputs where stylistically similar to an AI assistant, and inputs were diverse.

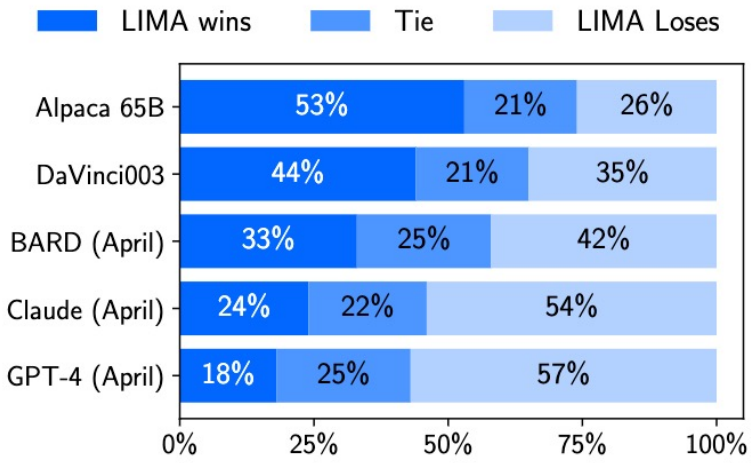


Figure 1: Human preference evaluation, comparing LIMA to 5 different baselines across 300 test prompts.

*"… 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."*

https://arxiv.org/pdf/2305.11206; Year: 2023

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.
```

s_1

$R(s_1) = 8.0$

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

s_2

$R(s_2) = 1.2$

- 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!

s_1

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

$R(s_1) = 8.0$

s_2

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

$R(s_2) = 1.2$

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!

s_1: `An earthquake hit San Francisco. There was minor property damage, but no injuries.`

$R(s_1) = 8.0$

s_2: `The Bay Area has good weather but is prone to earthquakes and wildfires.`

$R(s_2) = 1.2$

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

$$s_3$$

$$R(s_3) = \quad 4.1? \quad 6.6? \quad 3.2?$$

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.`	$>$	`A 4.2 magnitude earthquake hit San Francisco, resulting in massive damage.`	$>$	`The Bay Area has good weather but is prone to earthquakes and wildfires.`
s_1		s_3		s_2

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

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

"winning" sample — s^w

"losing" sample — s^l

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 θ 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 θ 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:

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

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:

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

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

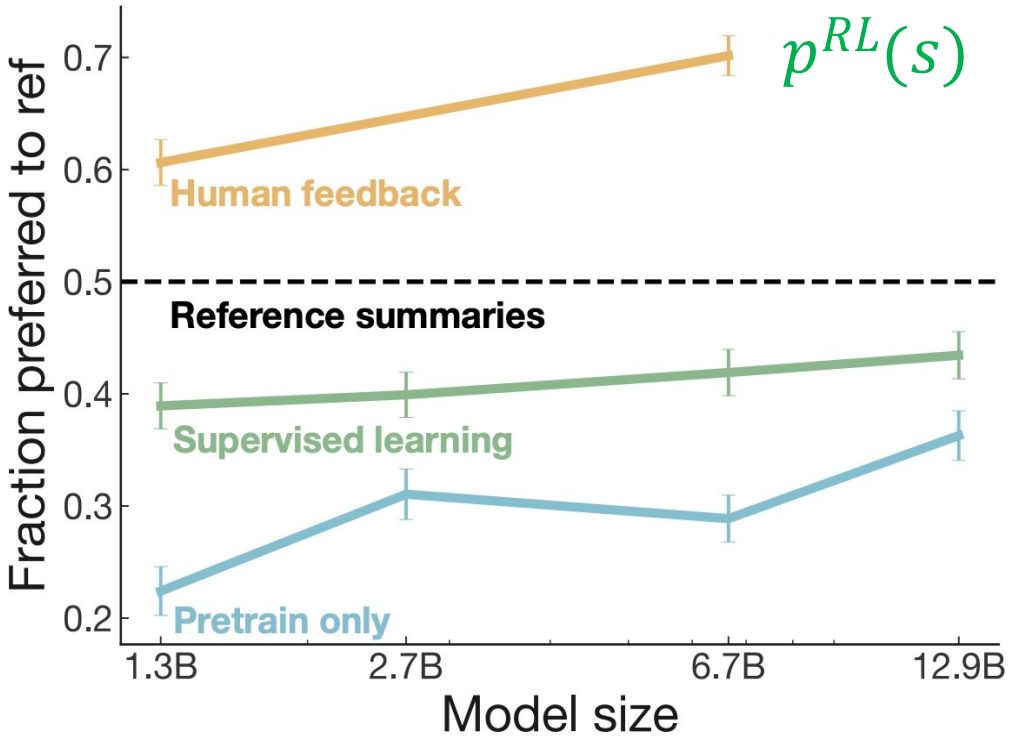
This is a penalty which prevents us from diverging too far from the pretrained model. In expectation, it is known as the **Kullback-Leibler** (**KL**) divergence between $p_\theta^{RL}(s)$ and $p^{PT}(s)$

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!

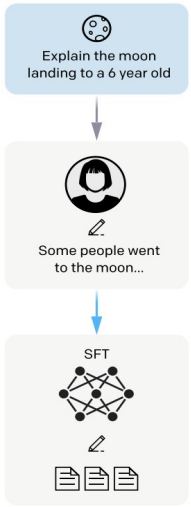
Step 1

Collect demonstration data, and train a supervised policy.

A prompt is sampled from our prompt dataset.

A labeler demonstrates the desired output behavior.

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



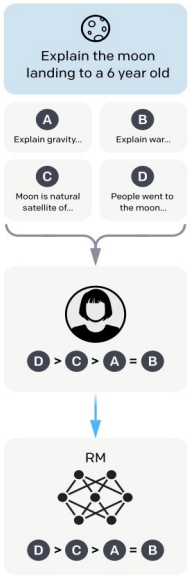
Step 2

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

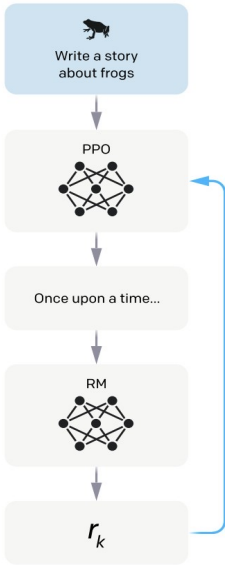
Optimize a policy against the reward model using reinforcement learning.

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.



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 $RM_\phi(x, y)$:

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

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

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

- The final DPO loss function is:

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

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

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

BUSINESS • TECHNOLOGY

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

15 MINUTE READ

NIAMH ROWE BUSINESS 15.10.2023 08:00 AM

Millions of Workers Are Training AI Models for Pennies

From the Philippines to Colombia, low-paid workers label training data for AI models used by the likes of Amazon, Facebook, Google, and Microsoft.

Oskarina Vero Fuentes with her dog. COURTESY OF OSKARINA VERO FUENTES

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

By Rebecca Tan and Regine Cabato

August 28, 2023 at 2:00 a.m. EDT

“RLHF produces more performant models in-distribution and out-of-distribution, but at the cost of lower output diversity …”

https://openreview.net/pdf?id=PXD3FAVHJT

"RLHF produces more performant models in-distribution and out-of-distribution, but at the cost of lower output diversity …"

https://openreview.net/pdf?id=PXD3FAVHJT

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 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

https://www.anthropic.com/news/political-even-handedness

"RLHF produces more performant models in-distribution and out-of-distribution, but at the cost of lower output diversity …"

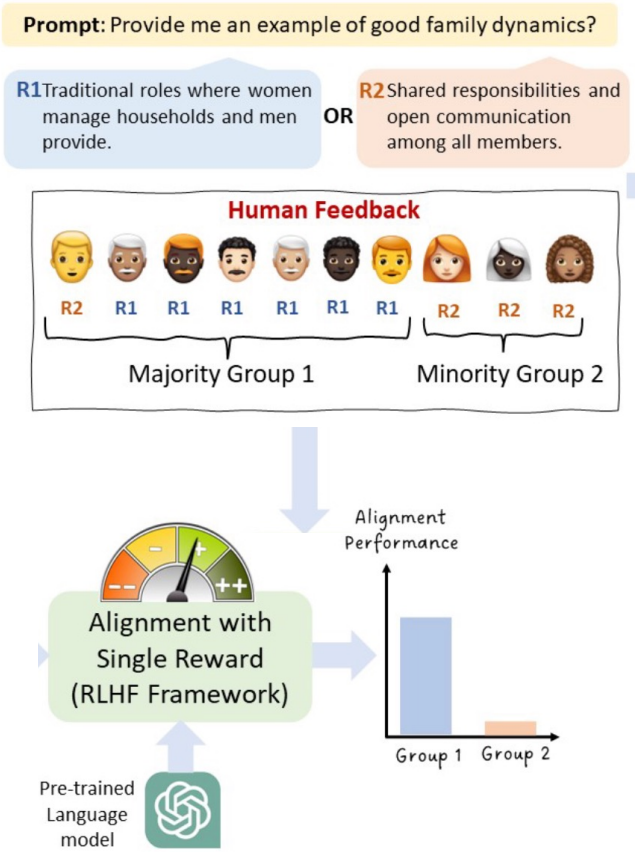
https://openreview.net/pdf?id=PXD3FAVHJT

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 reinforce demographic or political biases present in the labeler pool

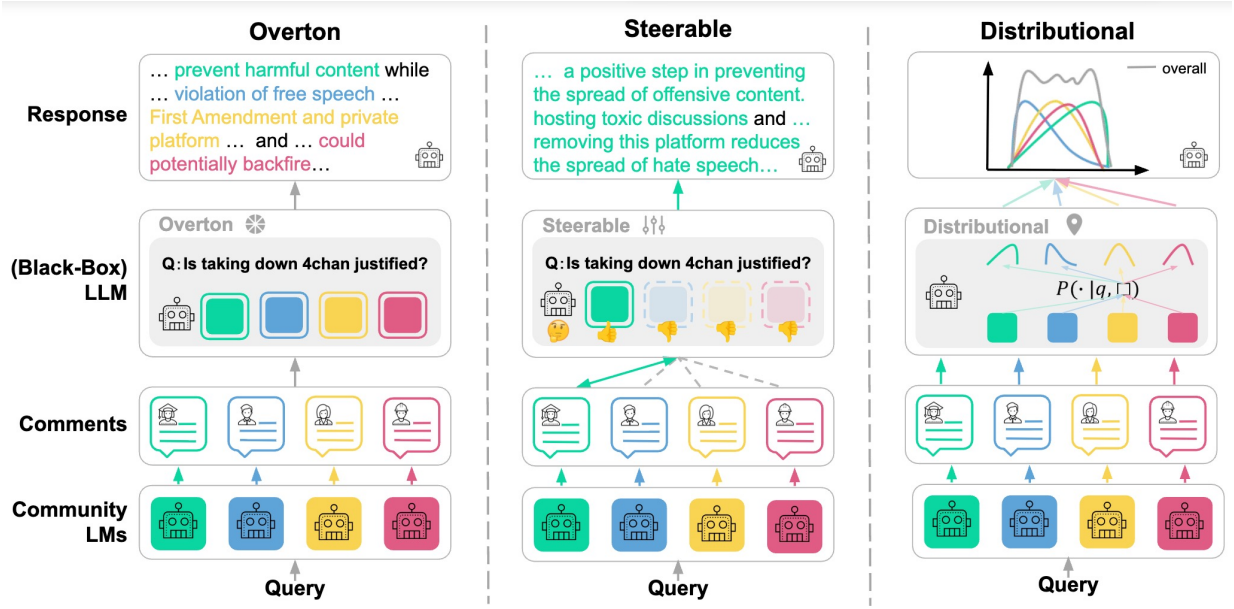
Some Mitigation Strategies:
- **Constitutional AI**: Using explicit principles rather than relying solely on crowdworker preferences
- **Character training**: Directly training models on specific neutrality traits during alignment
- **System prompts**: Providing explicit instructions for even-handed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



https://arxiv.org/abs/2402.08925 Propose a mixture of rewards

https://www.anthropic.com/news/political-even-handedness

Pluralistic Alignment (post-hoc)



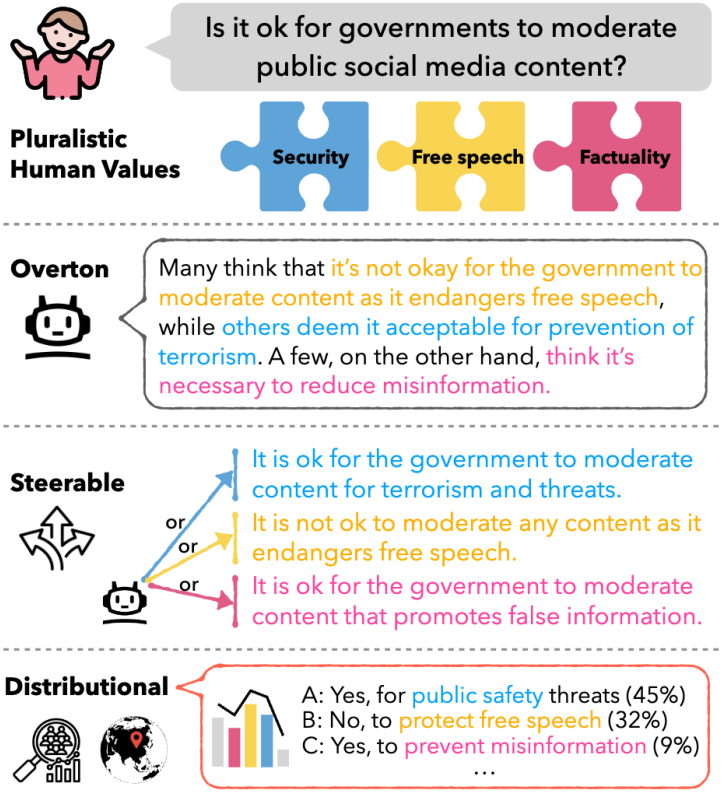


Figure 1. Three kinds of pluralism in models.

https://aclanthology.org/2024.emnlp-main.240

https://arxiv.org/pdf/2402.05070

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)	**40.00**	**58.59**	**91.60**	**63.40**	60.40
LRM + DirectRefusal	13.33	35.35	48.80	32.49	**0.80**

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]
Equall
MICS - CentraleSupelec

Telmo Pires [equall]
Equall

Malik Boudiaf [equall]
Equall

Rui Melo [equall]
Equall

Dominic Culver [equall]
Equall

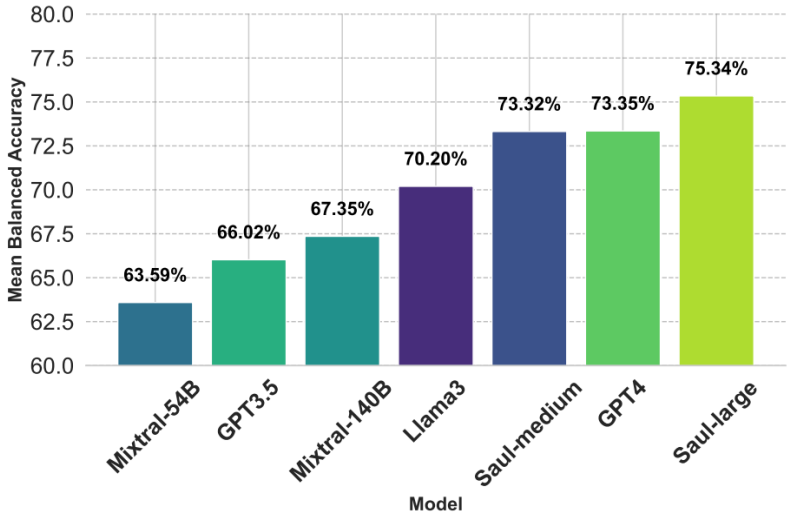
Sofia Morgado [equall]
Equall

Etienne Malaboeuf
CINES

Gabriel Hautreux
CINES

Johanne Charpentier
CINES

Michael Desa [equall]
Equall

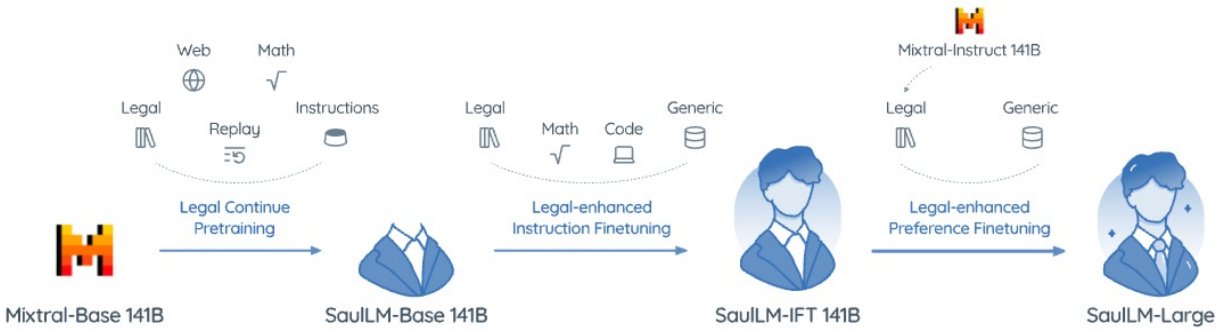


https://arxiv.org/pdf/2407.19584

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
Total	520

https://arxiv.org/pdf/2407.19584

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.

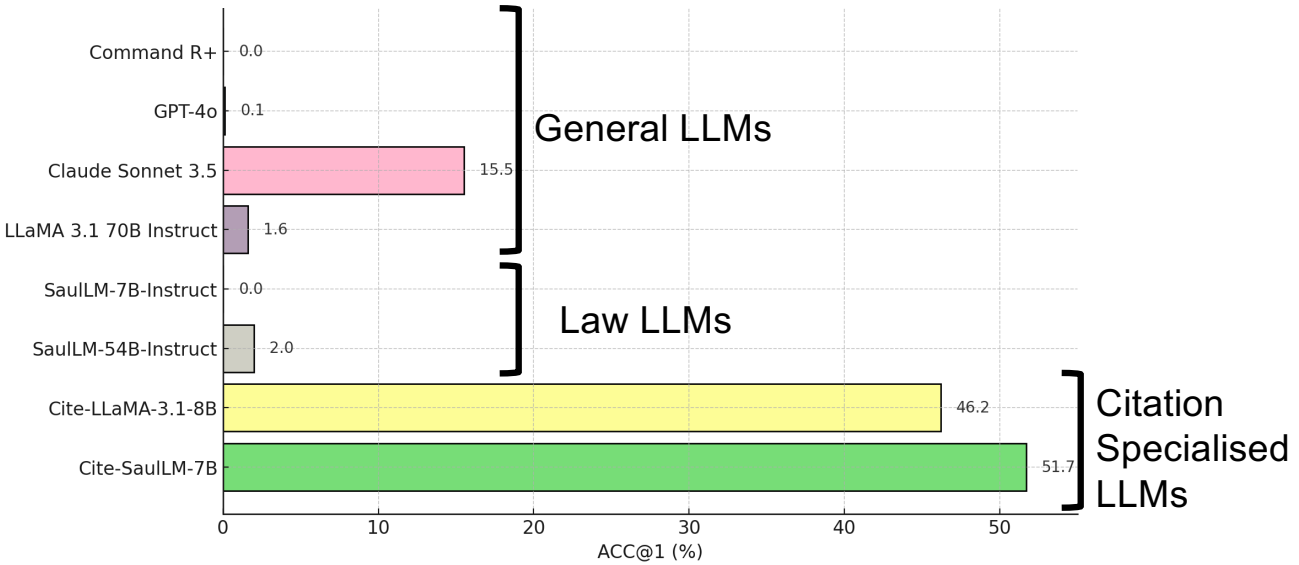
Legal-specific – Synthetic legal scenarios; responses are scored for factuality, relevance and logic by Mixtral-142B-Instruct to create accepted/rejected pairs for Direct Preference Optimisation (DPO).

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