10-423/623: Generative AI Lecture 11 – Reinforcement Learning with Human Feedback

Henry Chai & Matt Gormley

10/2/24

Front Matter

- Announcements:
  - HW2 released ~~9/23~~ 9/24, due 10/7 at 11:59 PM

Recall: Few-shot Learning with LLMs

- Suppose you have…
  1. a small labelled dataset (i.e., a few-shot setting), $\mathcal{D}$
  2. a very large pre-trained language model
- There are two ways to "learn":

A. Supervised fine-tuning i.e., updating the LLM's parameters using
  1. a standard supervised objective
  2. backpropagation to compute gradients
  3. your favorite optimizer (e.g., Adam)

- **Con**: backpropagation requires ~3x the memory and computation time as the forward computation
- **Con**: you might not have access to the model parameters

Recall: Few-shot Learning with LLMs

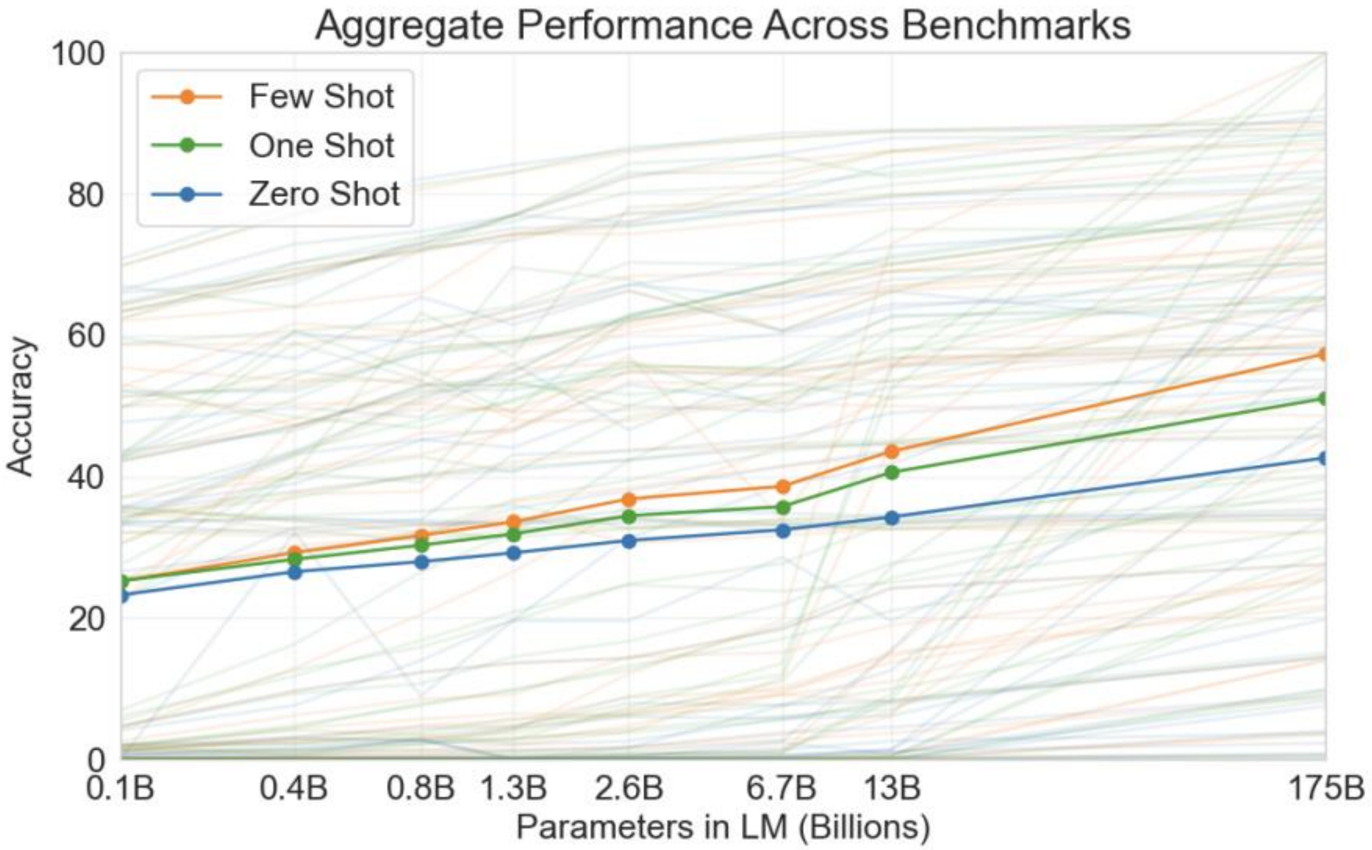
- Suppose you have…
  1. a small labelled dataset (i.e., a few-shot setting), $\mathcal{D}$
  2. a very large pre-trained language model
- There are two ways to "learn":

B. In-context learning i.e., feeding the training dataset to the LLM as a prompt and taking the output as a prediction
   - the LLM (hopefully) infers patterns in the training dataset during inference (i.e., decoding)

- **Pro**: no backpropagation required and only one pass through the training dataset *per test example*
- **Pro**: does not require access to the model parameters, only API access to the model itself
- **Con**: the prompt may be very long and Transformer LMs require $O(N^2)$ time/space where N = length of context

Recall: Few-shot Learning via In-context Learning with LLMs

- Standard setup: a set of input/output pairs from a training dataset are presented in sequence to an LLM, typically along with a plain-text task description



Source: https://arxiv.org/pdf/2005.14165

Recall: Few-shot In-context Learning with LLMs

- In-context learning is surprisingly *sensitive* to…
  1. the order the training examples are presented in
  2. label imbalance (e.g. # of positive vs. # of negative)
  3. the number of unique labels in the training dataset
- In-context learning is surprisingly *insensitive* to…
  1. the correctness of the labels!
  2. the amount of training data used in the prompt!

So why does *this* work? Why is it better than zero-shot learning?

- In-context learning is surprisingly *sensitive* to…
  1. the order the training examples are presented in
  2. label imbalance (e.g. # of positive vs. # of negative)
  3. the number of unique labels in the training dataset
- In-context learning is surprisingly *insensitive* to…
  1. the correctness of the labels!
  2. the amount of training data used in the prompt!

Few-shot In-context Learning with LLMs

- Min et al. (2022) identified four meaningful factors:

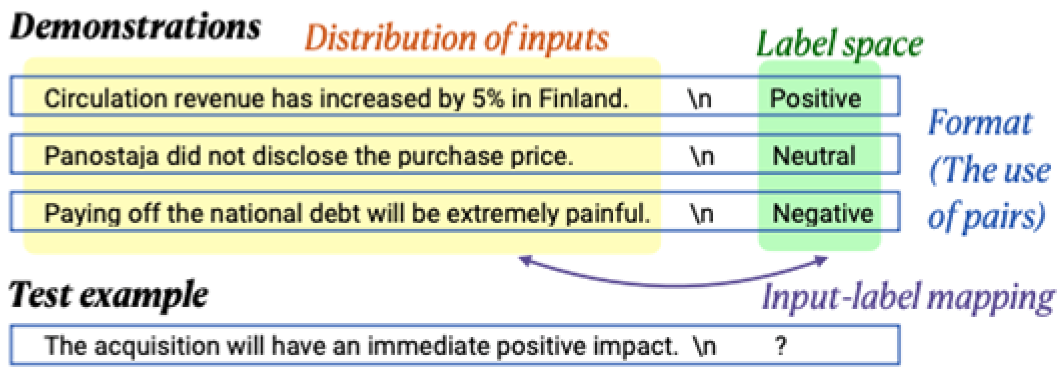


Figure 7: Four different aspects in the demonstrations: the input-label mapping, the distribution of the input text, the label space, and the use of input-label pairing as the format of the demonstrations.

- Another potentially meaningful aspect of in-context learning: what *exactly* are we asking the LLM?

Source: http://arxiv.org/abs/2202.12837

Prompt Engineering

- Not all prompts are equally good!
- Example: zero-shot news article classification using OPT-175B on the AG News dataset

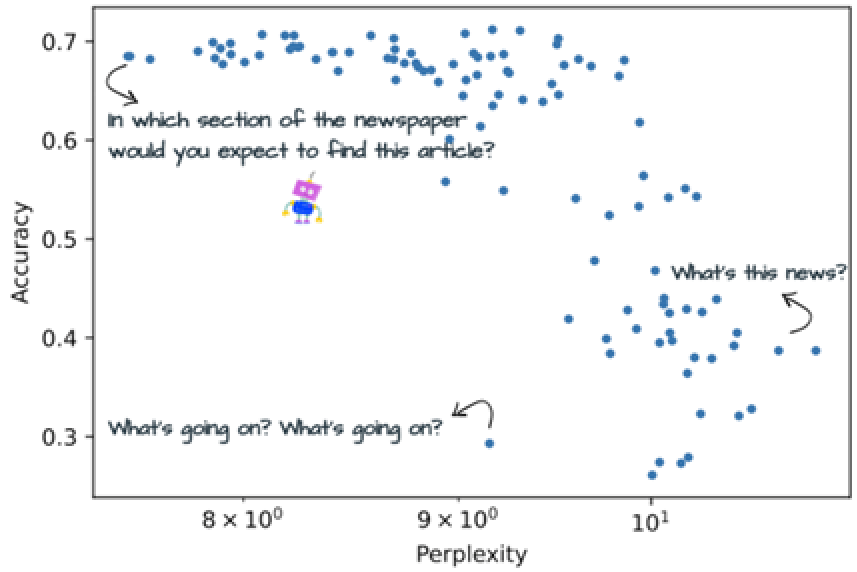
Prompt	Accuracy
What is this piece of news regarding?	40.9
What is this article about?	52.4
What is the best way to describe this article?	68.2
What is the most accurate label for this news article?	71.2

- What affects the accuracy associated with using a prompt?
- One potential answer: how likely the prompt is under the learned model's implied distribution over sequences

Source: https://arxiv.org/pdf/2212.04037

Prompt Engineering

- Not all prompts are equally good!
- Example: zero-shot news article classification using OPT-175B on the AG News dataset



- Perplexity is the exponentiated average negative log-likelihood of a sequence
- Lower perplexity = higher likelihood

Source: https://arxiv.org/pdf/2212.04037

Prompt Engineering

- Not all prompts are equally good!
- Example: zero-shot news article classification using OPT-175B on the AG News dataset

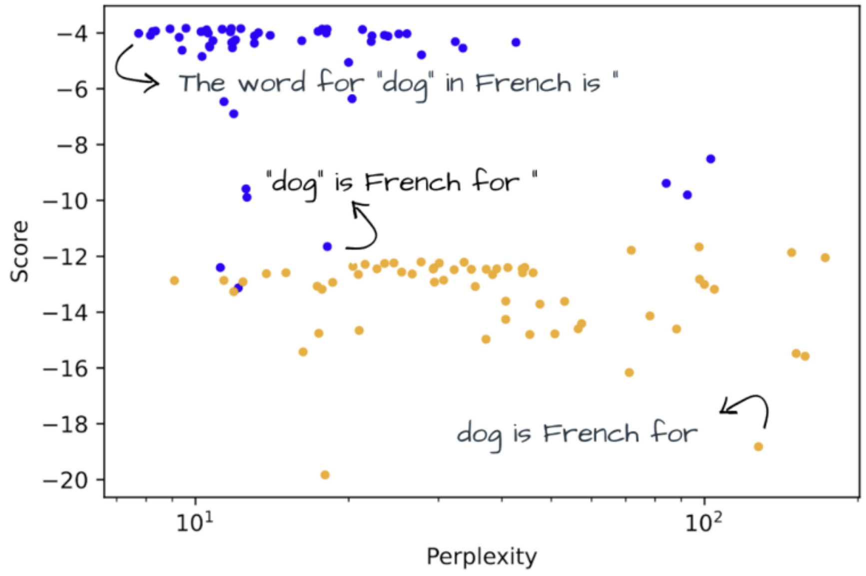


Figure 2: Score of correct label vs. perplexity for the word-level translation task in French with OPT 175B. The x axis is in log scale. The blue points stand for prompts with quotation marks for the words, while the yellow points are of prompts without quotation marks.

- Perplexity is the exponentiated average negative log-likelihood of a sequence
- Lower perplexity = higher likelihood

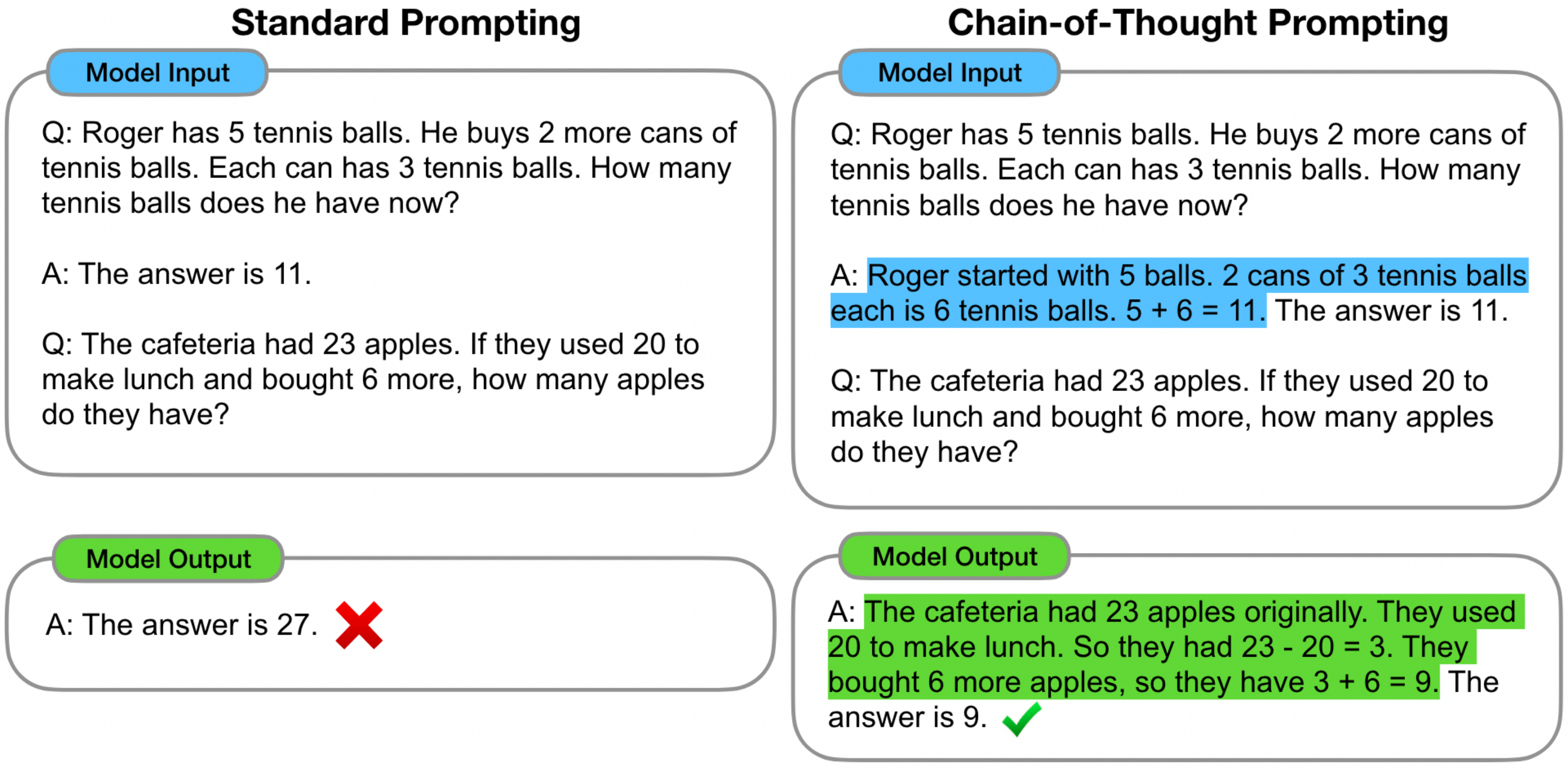
Source: https://arxiv.org/pdf/2212.04037

Learning to Prompt

- Some ways of *learning* better prompts for your task:
    1. Prompt paraphrasing – programmatically generate and test many different prompts from a paraphrase model, then pick the one that works best
    2. Gradient-based search – use optimization to search for the discrete representation of the prompt that makes the desired output most likely
    3. Prompt tuning – directly optimize the embeddings that are input into the LLM, without bothering to construct a discrete representation of the prompt

Chain-of-Thought Prompting

- Insight: asking an LLM to reason about its answer can improve its in-context performance
- Chain-of-thought prompting provides examples of reasoning in the in-context training examples



Source: http://arxiv.org/abs/2201.11903

Chain-of-Thought Prompting

- Insight: asking an LLM to reason about its answer can improve its in-context performance
- Chain-of-thought prompting provides examples of reasoning in the in-context training examples

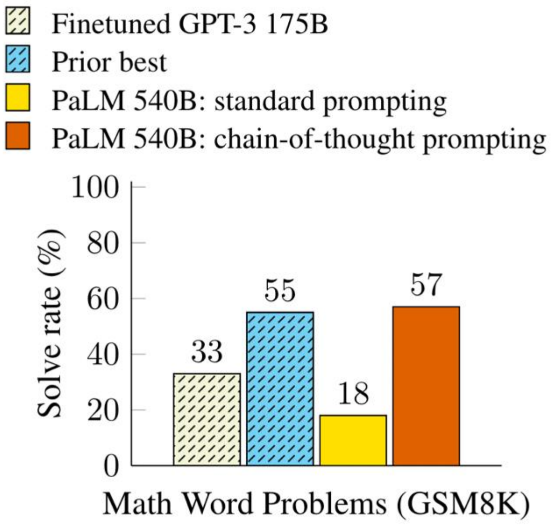
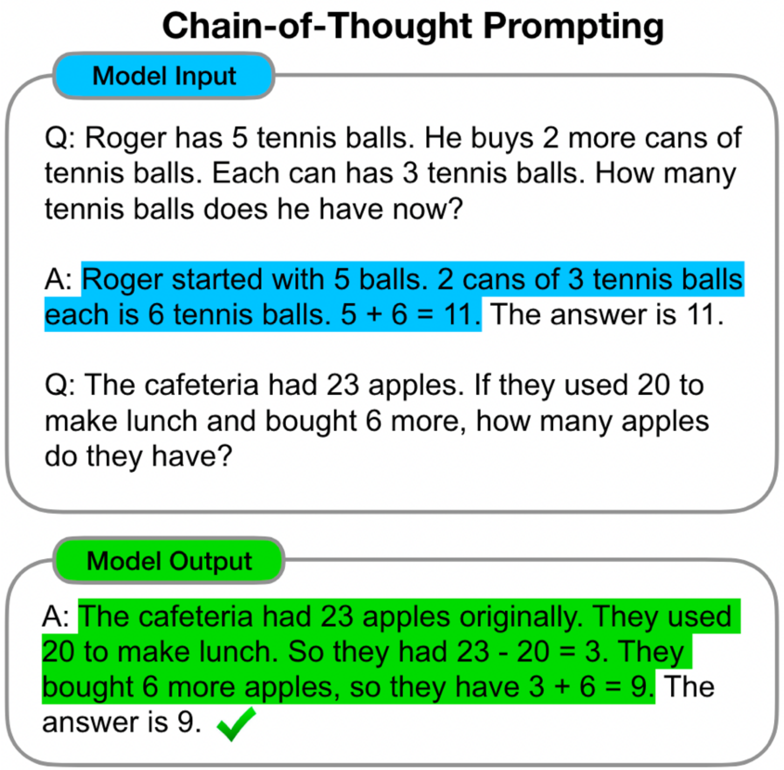


Figure 2: PaLM 540B uses chain-of-thought prompting to achieve new state-of-the-art performance on the GSM8K benchmark of math word problems. Finetuned GPT-3 and prior best are from Cobbe et al. (2021).



Source: http://arxiv.org/abs/2201.11903

Chain-of-Thought Prompting

- Insight: asking an LLM to reason about its answer can improve its in-context performance
- Chain-of-thought prompting provides examples of reasoning in the in-context training examples

(a) Few-shot

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: 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:

(Output) The answer is 8. X

(b) Few-shot-CoT

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: 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:

(Output) The juggler can juggle 16 balls. Half of the balls are golf balls. So there are 16 / 2 = 8 golf balls. Half of the golf balls are blue. So there are 8 / 2 = 4 blue golf balls. The answer is 4. ✓

(c) Zero-shot

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: The answer (arabic numerals) is

(Output) 8 X

Source: https://arxiv.org/pdf/2205.11916.pdf

Chain-of-Thought Prompting

- Insight: asking an LLM to reason about its answer can improve its in-context performance
- Chain-of-thought prompting provides examples of reasoning in the in-context training examples

	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

Source: https://arxiv.org/pdf/2205.11916.pdf

Chain-of-Thought Prompting

- Insight: asking an LLM to reason about its answer can improve its in-context performance
- Chain-of-thought prompting provides examples of reasoning in the in-context training examples

September 12, 2024

Learning to Reason with LLMs

We are introducing OpenAI o1, a new large language model trained with reinforcement learning to perform complex reasoning. o1 thinks before it answers—it can produce a long internal chain of thought before responding to the user.

Source: https://openai.com/index/learning-to-reason-with-llms/

Recall: Zero-shot Learning with LLMs

- Summarization

Prompt:

Story: Jason listened to the weather and heard it was going to be sunny. He thought the kids might like to go swimming. He gathered up the swimsuits, towels and sunscreen. Jason and the kids got into the truck and drove to the beach. They spent the next 2 hours playing and splashing in the surf.

One-sentence Summary:

Model output (Llama-2-70B):

Jason took the kids swimming at the beach after hearing the weather forecast, gathering necessary items and driving there.

Model output (Llama-2-7B Chat):

Sure! Here is a one-sentence summary of the story:

Jason and the kids went on a fun-filled day at the beach, playing and splashing in the surf after gathering all the necessary items, including swimsuits, towels, and sunscreen.

Source: https://arxiv.org/pdf/1910.00998.pdf

Instruction Fine-Tuning

- LLMs are trained to reduce the perplexity of its training corpus i.e., it's good at completing __________
- Issue: an interactive agent (e.g., a chatbot) can't just predict what comes next, it needs to *behave* in a certain way (e.g., conversationally)
  - We need to *align* the LLM with the expectations of a human user for a given task
- Idea: build a "chatbot" training dataset and use it to fine-tune the LLM
  - This approach is called instruction fine-tuning, chat fine-tuning, alignment, behavioral fine-tuning, etc…

So how could we build such a dataset? Where can we get…

- Prompts?
- Response?

- LLMs are trained to reduce the perplexity of its training corpus i.e., it’s good at completing __________
- Issue: an interactive agent (e.g., a chatbot) can’t just predict what comes next, it needs to *behave* in a certain way (e.g., conversationally)
  - We need to *align* the LLM with the expectations of a human user for a given task
- Idea: build a **“chatbot” training dataset** and use it to fine-tune the LLM
  - This approach is called instruction fine-tuning, chat fine-tuning, alignment, behavioral fine-tuning, etc…

Instruction Fine-Tuning Dataset: InstructGPT

- Consists of 13k prompt/response pairs
- Human labelers wrote both an instruction prompt and a demonstration response
- Some prompts were taken from early users of the OpenAI API, and labelers wrote the demo response
- All data is *closed source*

Table 1: Distribution of use case categories from our API prompt dataset.

Use-case	**(%)**
Generation	45.6%
Open QA	12.4%
Brainstorming	11.2%
Chat	8.4%
Rewrite	6.6%
Summarization	4.2%
Classification	3.5%
Other	3.5%
Closed QA	2.6%
Extract	1.9%

Source: https://arxiv.org/pdf/2203.02155

Instruction Fine-Tuning Dataset: Dolly

- Consists of 15k prompt/response pairs
- Employees of DataBricks wrote all the prompt/response pairs

We set up a contest, where the top 20 labelers would get a big award. We also outlined 7 very specific tasks:

- Open Q&A: For instance, "Why do people like comedy movies?" or "What is the capital of France?" In some cases, there's not a correct answer, and in others, it requires drawing on knowledge of the world at large.
- Closed Q&A: These are questions that can be answered using only the information contained in a passage of reference text. For instance, given a paragraph from Wikipedia on the atom, one might ask, "What is the ratio between protons and neutrons in the nucleus?"
- Extract information from Wikipedia: Here an annotator would copy a paragraph from Wikipedia and extract entities or other factual information such as weights or measurements from the passage.
- Summarize information from Wikipedia: For this, annotators provided a passage from Wikipedia and were asked to distill it to a short summary.
- Brainstorming: This task asked for open-ended ideation and an associated list of possible options. For instance, "What are some fun activities I can do with my friends this weekend?".
- Classification: For this task, annotators were asked to make judgments about class membership (e.g. are the items in a list animals, minerals or vegetables) or to judge the properties of a short passage of text, such as the sentiment of a movie review.
- Creative writing: This task would include things like writing a poem or a love letter.

- All data is *open source!*

Source: https://www.databricks.com/blog/2023/04/12/dolly-first-open-commercially-viable-instruction-tuned-llm

Instruction Fine-Tuning Dataset: Dolly

Category	Instruction	Response
General Q&A	What is best Playstation or Xbox?	Playstation and Xbox are two competing game consoles, where which is "best" will depend on the preferences of the gamer. From a graphics and performance perspective they are comparable, with Sony's Playstation technically leading, though most gamers would be unlikely to notice a significant difference. From a game library perspective, both consoles have their respective first-party titles that appeal to different demographics. For example, Microsoft has seen great popularity due to popular title Halo being exclusively available on the console. On the other hand, Playstation has popular exclusive titles such as God of War and Gran Turismo. Ultimately the decision of which console is best will likely come down to which games titles the purchaser prefers.

Source: https://huggingface.co/datasets/databricks/databricks-dolly-15k

Instruction Fine-Tuning Dataset: Dolly

Category	Instruction	Response
Open Q&A	What individual has won the most Olympic gold medals in the history of the games?	Michael Phelps has won the most gold medals of all time with 23 golds.

Source: https://huggingface.co/datasets/databricks/databricks-dolly-15k

Instruction Fine-Tuning Dataset: Dolly

Category	Instruction	Context	Response
Closed Q&A	When was the first Reading railway station opened?	Reading railway station is a major transport hub in Reading, Berkshire, England. It is on the northern edge of the town centre, near the main retail and commercial areas and the River Thames, 36 miles (58 km) from London Paddington. The first Reading station was opened on 30 March 1840 as the temporary western terminus of the original line of the Great Western Railway (GWR). Reading is the ninth-busiest station in the UK outside London and the second busiest interchange station outside London with over 3.8 million passengers changing trains at the station annually.	The first Reading railway station was opened on the 30th of March, 1840.

Source: https://huggingface.co/datasets/databricks/databricks-dolly-15k

Instruction Fine-Tuning Dataset: Dolly

Category	Instruction	Context	Response
Information Extraction	Extract all of the dates mentioned in this paragraph and list them using bullets in the format {Date} - {Description}	Daniel Leavitt was born November 16, 1813, at Rye, New Hampshire, the son of Benning Leavitt, influential businessman, state senator, county commissioner and later Chicopee Selectman, and his wife Olive (Jenness) Leavitt. Daniel Leavitt married in 1838 at West Springfield, Massachusetts, Ruth Jeannette Ball. They had three children.	• November 16, 1813 - This is when Daniel Leavitt was born • 1838 - This is when Daniel Leavitt married his wife Ruth Jeannette Ball.

Source: https://huggingface.co/datasets/databricks/databricks-dolly-15k

Instruction Fine-Tuning Dataset: Dolly

Category	Instruction	Response
Brainstorming	What are some unique curtain tie backs that you can make yourself?	There are many items you can use to make a curtain tie back. Some ideas for this include a chain, a circle and pin, jute rope, wooden beaded rope, a necklack or bracelet, a door knob, a length of a leather belt, macrame rope, or a string of artificial flowers.
Classification	Identify which instrument is string or percussion: Cantaro, Gudok	Gudok is string, Cantaro is percussion.

Source: https://huggingface.co/datasets/databricks/databricks-dolly-15k

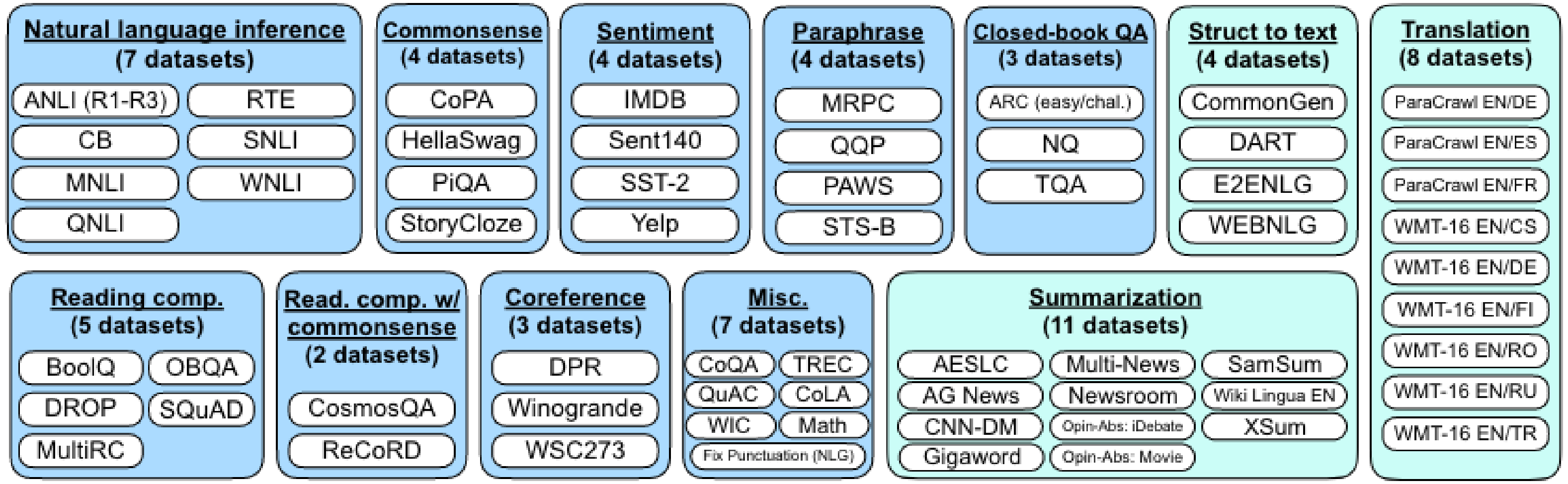
Instruction Fine-Tuning Dataset: Dolly

Category	Instruction	Response
Creative Writing	Write a haiku about how much I love pickles.	Caressed by sweet brine, you are crispy and crunchy. Oh, won't you be mine?
Creative Writing	What should I do on a free afternoon in San Francisco?	You can visit and take photos at San Francisco's landmarks such as Pier 39 or Golden Gate Bridge. If you enjoy walking outdoors, there are several large parks in the city. You can visit Golden Gate Park and the various attractions and museums located in the park. The Presidio is also a popular park where you can get some of the best views of Golden Gate Bridge.

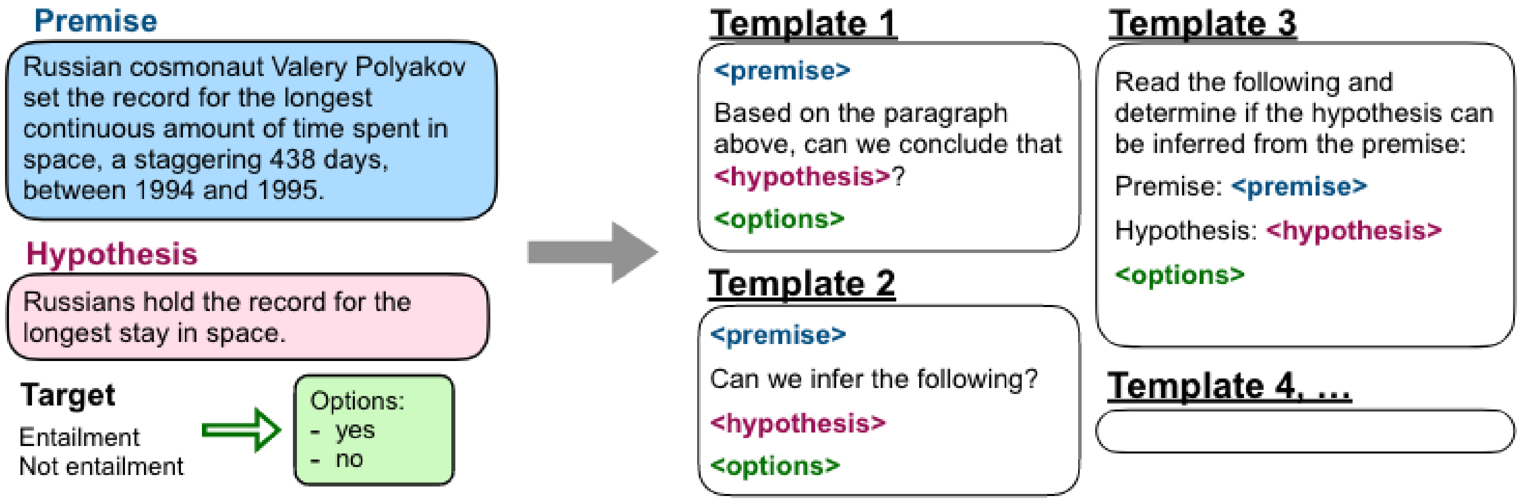
Source: https://huggingface.co/datasets/databricks/databricks-dolly-15k

Instruction Fine-Tuning Dataset: Flan

- **Key idea:** *programmatically* build instruction fine-tuning examples from existing NLP tasks (12 tasks, 62 datasets)



- For each NLP task/dataset, Flan created 10 templates e.g.,



Source: https://arxiv.org/pdf/2109.01652

Instruction Fine-Tuning Datasets

Release	Collection	Model	Base	Size	Public?	Prompt Types	Tasks in Flan	# Exs	Methods
		Model Details				**Data Collection & Training Details**			
2020 05	UnifiedQA	UnifiedQA	RoBerta	110-340M	P	ZS	46 / 46	750k	
2021 04	CrossFit	BART-CrossFit	BART	140M	NP	FS	115 / 159	71.M	
2021 04	Natural Inst v1.0	Gen. BART	BART	140M	NP	ZS / FS	61 / 61	620k	+ Detailed k-shot Prompts
2021 09	Flan 2021	Flan-LaMDA	LaMDA	137B	NP	ZS / FS	62 / 62	4.4M	+ Template Variety
2021 10	P3	T0, T0+, T0++	T5-LM	3-11B	P	ZS	62 / 62	12M	+ Template Variety + Input Inversion
2021 10	MetaICL	MetaICL	GPT-2	770M	P	FS	100 / 142	3.5M	+ Input Inversion + Noisy Channel Opt
2021 11	ExMix	ExT5	T5	220M-11B	NP	ZS	72 / 107	500k	+ With Pretraining
2022 04	Super-Natural Inst.	*Tk*-Instruct	T5-LM, mT5	11-13B	P	ZS / FS	1556 / 1613	5M	+ Detailed k-shot Prompts + Multilingual
2022 10	GLM	GLM-130B	GLM	130B	P	FS	65 / 77	12M	+ With Pretraining + Bilingual (en, zh-cn)
2022 11	xP3	BLOOMz, mT0	BLOOM, mT5	13-176B	P	ZS	53 / 71	81M	+ Massively Multilingual
2022 12	Unnatural Inst.†	T5-LM-Unnat. Inst.	T5-LM	11B	NP	ZS	~20 / 117	64k	+ Synthetic Data
2022 12	Self-Instruct†	GPT-3 Self Inst.	GPT-3	175B	NP	ZS	Unknown	82k	+ Synthetic Data + Knowledge Distillation
2022 12	OPT-IML Bench†	OPT-IML	OPT	30-175B	P	ZS + FS CoT	~2067 / 2207	18M	+ Template Variety + Input Inversion + Multilingual
2022 10	Flan 2022 (ours)	Flan-T5, Flan-PaLM	T5-LM, PaLM	10M-540B	P NP	ZS + FS CoT	1836	15M	+ Template Variety + Input Inversion + Multilingual

Figure 2: A **Timeline of Public Instruction Tuning Collections** specifies the collection release date, detailed information on the finetuned models (the base model, their size, and whether the model itself is Public (P) or Not Public (NP)), what prompt specification they were trained for (zero-shot, few-shot, or Chain-of-Thought), the number of tasks contained in the Flan 2022 Collection (released with this work), and core methodological contributions in each work.
Note that the number of tasks and of examples vary under different assumptions and so are estimates. For instance, the definition of "task" and "task category" vary by work, and are not easily simplified to one ontology. The reported counts for the number of tasks are reported using task definitions from the respective works.
† indicates concurrent work.

Source: https://arxiv.org/pdf/2301.13688

Instruction Fine-Tuning Models

Instruction fine-tuned LLMs	# Params	Base Model	Fine-tuning Trainset		
			Self-build	Dataset Name	Size
Instruct-GPT (Ouyang et al., 2022)	176B	GPT-3 (Brown et al., 2020b)	Yes	-	-
BLOOMZ (Muennighoff et al., 2022)[1]	176B	BLOOM (Scao et al., 2022)	No	xP3	-
FLAN-T5 (Chung et al., 2022)[2]	11B	T5 (Raffel et al., 2019)	No	FLAN 2021	-
Alpaca (Taori et al., 2023)[3]	7B	LLaMA (Touvron et al., 2023a)	Yes	-	52K
Vicuna (Chiang et al., 2023)[4]	13B	LLaMA (Touvron et al., 2023a)	Yes	-	70K
GPT-4-LLM (Peng et al., 2023)[5]	7B	LLaMA (Touvron et al., 2023a)	Yes	-	52K
Claude (Bai et al., 2022b)	-	-	Yes	-	-
WizardLM (Xu et al., 2023a)[6]	7B	LLaMA (Touvron et al., 2023a)	Yes	Evol-Instruct	70K
ChatGLM2 (Du et al., 2022)[7]	6B	GLM (Du et al., 2022)	Yes	-	1.1 Tokens
LIMA (Zhou et al., 2023)	65B	LLaMA (Touvron et al., 2023a)	Yes	-	1K
OPT-IML (Iyer et al., 2022)[8]	175B	OPT (Zhang et al., 2022a)	No	-	-
Dolly 2.0 (Conover et al., 2023)[9]	12B	Pythia (Biderman et al., 2023)	No	-	15K
Falcon-Instruct (Almazrouei et al., 2023a)[10]	40B	Falcon (Almazrouei et al., 2023b)	No	-	-
Guanaco (JosephusCheung, 2021)[11]	7B	LLaMA (Touvron et al., 2023a)	Yes	-	586K
Minotaur (Collective, 2023)[12]	15B	Starcoder Plus (Li et al., 2023f)	No	-	-
Nous-Hermes (NousResearch, 2023)[13]	13B	LLaMA (Touvron et al., 2023a)	No	-	300K+
TÜLU (Wang et al., 2023c)[14]	6.7B	OPT (Zhang et al., 2022a)	No	Mixed	-
YuLan-Chat (YuLan-Chat-Team, 2023)[15]	13B	LLaMA (Touvron et al., 2023a)	Yes	-	250K
MOSS (Tianxiang and Xipeng, 2023)[16]	16B	-	Yes	-	-
Airoboros (Durbin, 2023)[17]	13B	LLaMA (Touvron et al., 2023a)	Yes	-	-
UltraLM (Ding et al., 2023a)[18]	13B	LLaMA (Touvron et al., 2023a)	Yes	-	-

- Key takeaway: instruction fine-tuned models are often very effective at a *much* smaller scale than typical LLMs

Source: https://arxiv.org/pdf/2301.13688

Idea: What if we could get an LLM to generate its own fine-tuning dataset? How would we make sure the responses were "good"?

Instruction fine-tuned LLMs	# Params	Base Model	Fine-tuning Trainset		
			Self-build	Dataset Name	Size
Instruct-GPT (Ouyang et al., 2022)	176B	GPT-3 (Brown et al., 2020b)	Yes	-	-
BLOOMZ (Muennighoff et al., 2022)[1]	176B	BLOOM (Scao et al., 2022)	No	xP3	-
FLAN-T5 (Chung et al., 2022)[2]	11B	T5 (Raffel et al., 2019)	No	FLAN 2021	-
Alpaca (Taori et al., 2023)[3]	7B	LLaMA (Touvron et al., 2023a)	Yes	-	52K
Vicuna (Chiang et al., 2023)[4]	13B	LLaMA (Touvron et al., 2023a)	Yes	-	70K
GPT-4-LLM (Peng et al., 2023)[5]	7B	LLaMA (Touvron et al., 2023a)	Yes	-	52K
Claude (Bai et al., 2022b)	-	-	Yes	-	-
WizardLM (Xu et al., 2023a)[6]	7B	LLaMA (Touvron et al., 2023a)	Yes	Evol-Instruct	70K
ChatGLM2 (Du et al., 2022)[7]	6B	GLM (Du et al., 2022)	Yes	-	1.1 Tokens
LIMA (Zhou et al., 2023)	65B	LLaMA (Touvron et al., 2023a)	Yes	-	1K
OPT-IML (Iyer et al., 2022)[8]	175B	OPT (Zhang et al., 2022a)	No	-	-
Dolly 2.0 (Conover et al., 2023)[9]	12B	Pythia (Biderman et al., 2023)	No	-	15K
Falcon-Instruct (Almazrouei et al., 2023a)[10]	40B	Falcon (Almazrouei et al., 2023b)	No	-	-
Guanaco (JosephusCheung, 2021)[11]	7B	LLaMA (Touvron et al., 2023a)	Yes	-	586K
Minotaur (Collective, 2023)[12]	15B	Starcoder Plus (Li et al., 2023f)	No	-	-
Nous-Hermes (NousResearch, 2023)[13]	13B	LLaMA (Touvron et al., 2023a)	No	-	300K+
TÜLU (Wang et al., 2023c)[14]	6.7B	OPT (Zhang et al., 2022a)	No	Mixed	-
YuLan-Chat (YuLan-Chat-Team, 2023)[15]	13B	LLaMA (Touvron et al., 2023a)	Yes	-	250K
MOSS (Tianxiang and Xipeng, 2023)[16]	16B	-	Yes	-	-
Airoboros (Durbin, 2023)[17]	13B	LLaMA (Touvron et al., 2023a)	Yes	-	-
UltraLM (Ding et al., 2023a)[18]	13B	LLaMA (Touvron et al., 2023a)	Yes	-	-

- Key takeaway: instruction fine-tuned models are often very effective at a *much* smaller scale than typical LLMs

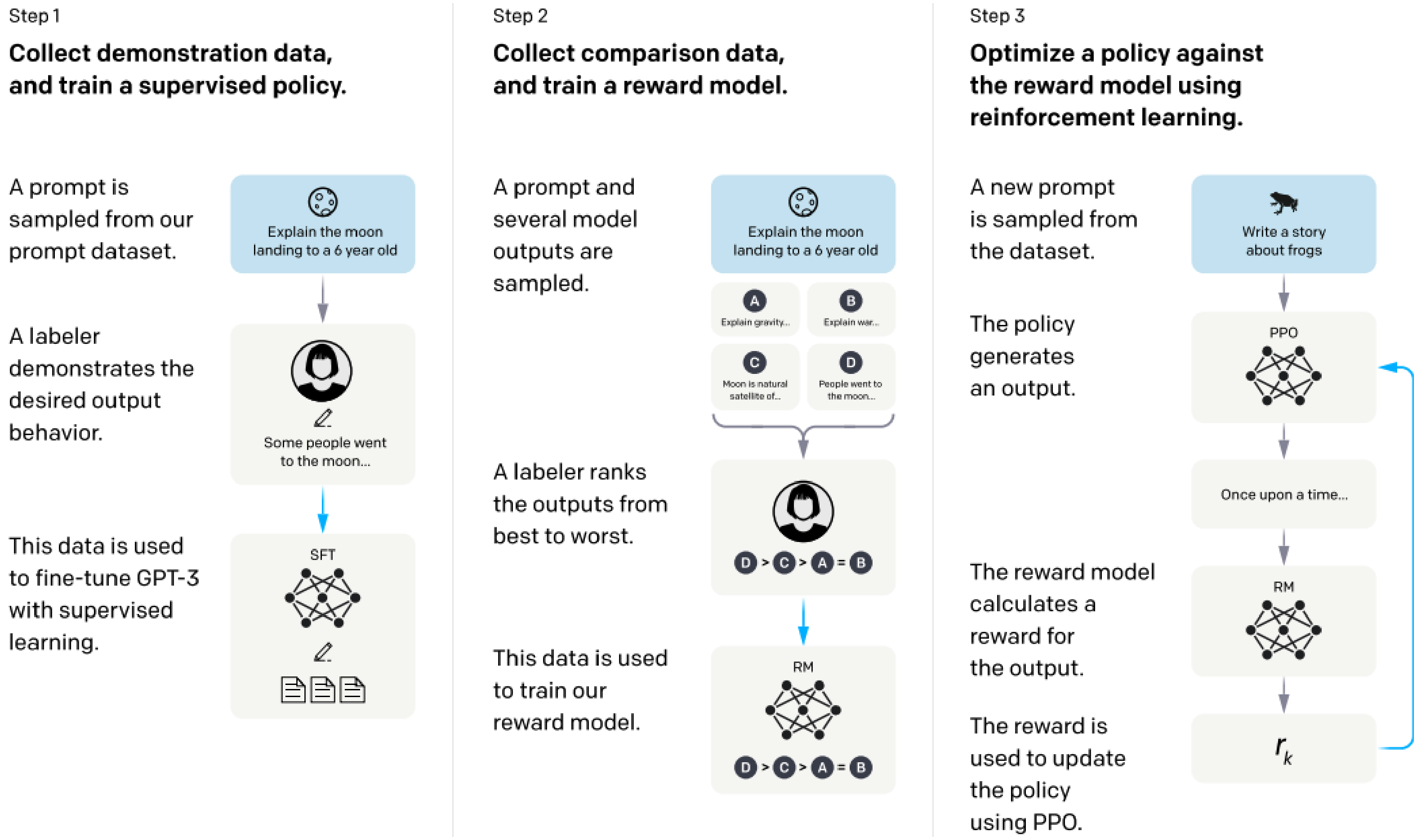
Source: https://arxiv.org/pdf/2301.13688

Reinforcement Learning from Human Feedback

- **InstructGPT** uses Reinforcement Learning with Human Feedback (RLHF) to *further fine-tune an already instruction fine-tuned version of GPT3*
  - "In human evaluations on our prompt distribution, outputs from the 1.3B parameter InstructGPT model are preferred to outputs from the 175B GPT-3, despite having 100x fewer parameters."

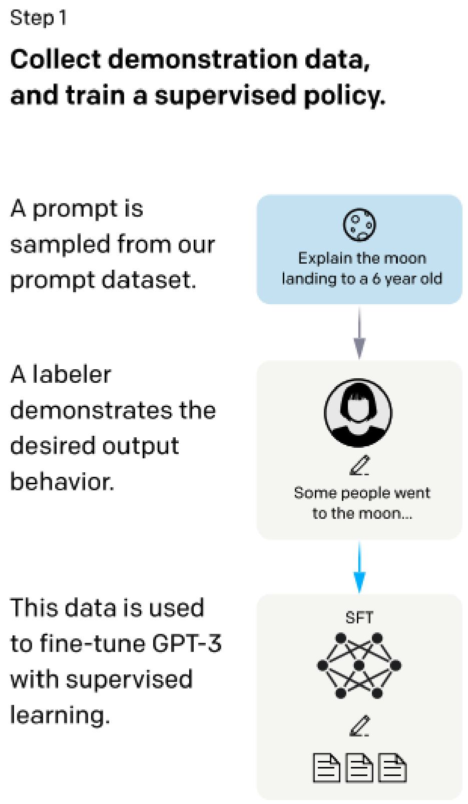
Source: https://arxiv.org/pdf/2203.02155

Reinforcement Learning from Human Feedback



Source: https://arxiv.org/pdf/2203.02155

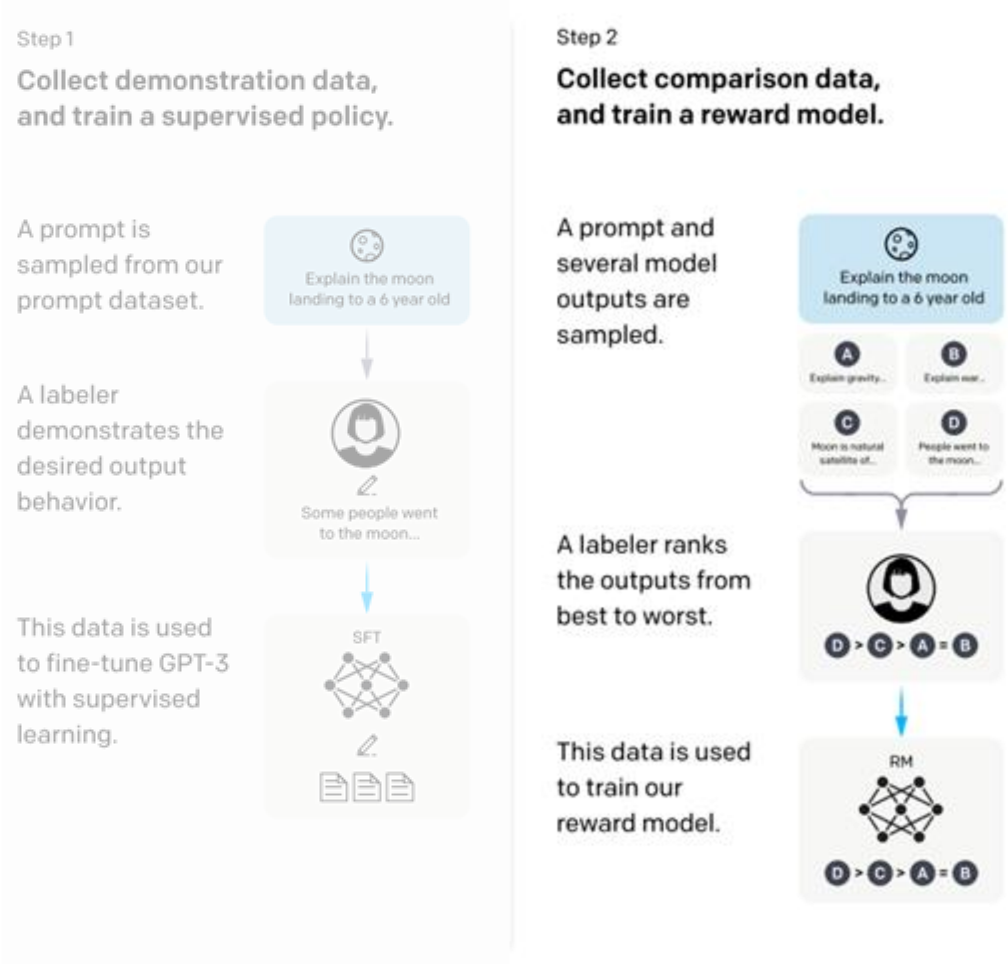
Reinforcement Learning from Human Feedback



- Step 1 performs instruction fine-tuning on 13k training examples
- This aligns the model behavior with what we would expect of a chat agent but the *diversity* of the responses might be limited because of the small (and potentially not representative) training data

Source: https://arxiv.org/pdf/2203.02155

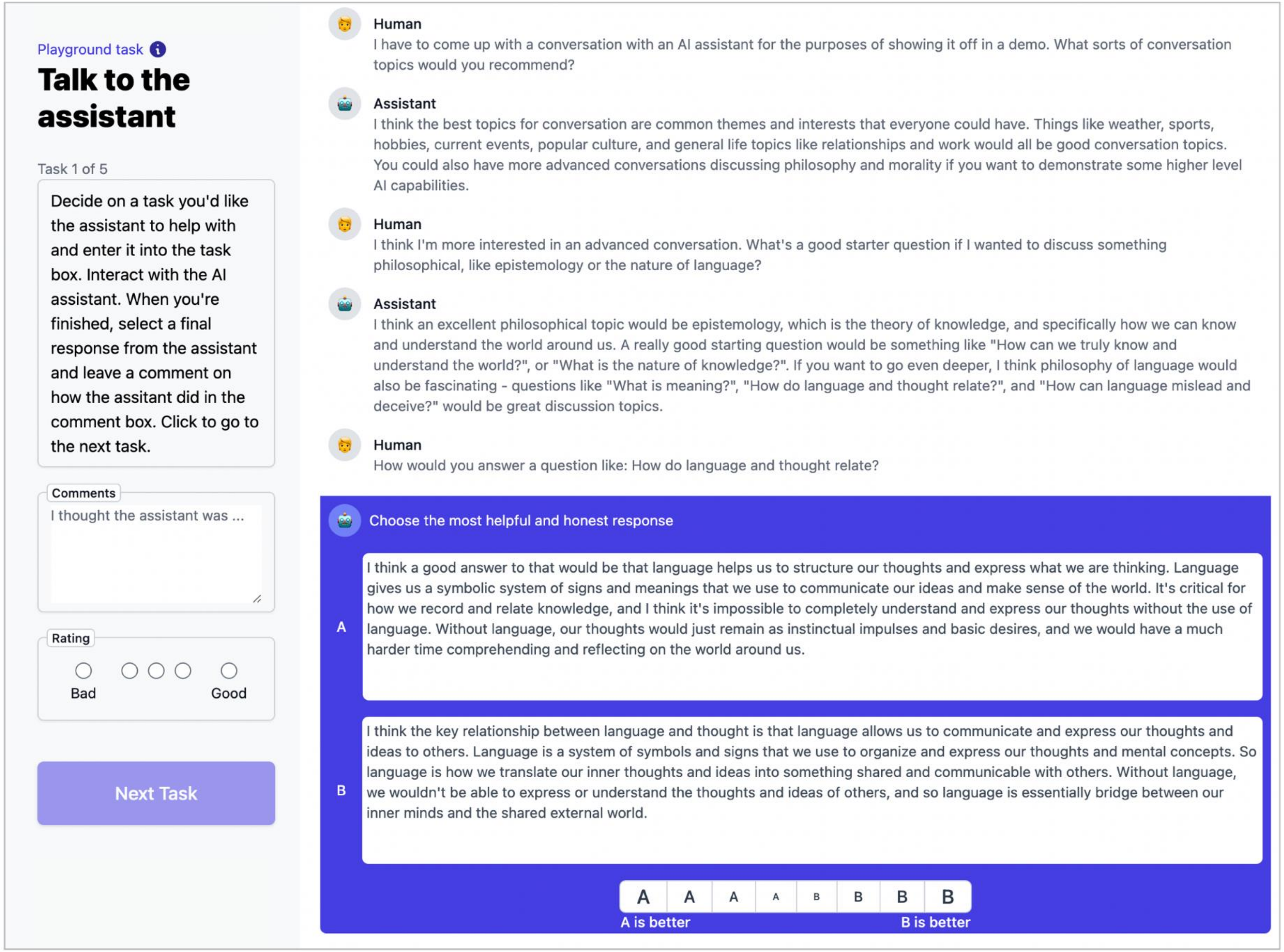
Reinforcement Learning from Human Feedback



- In Step 2, the model takes 33k prompts and samples $K \in \{4, \dots, 9\}$ responses from the instruction fine-tuned model for each one
- Then, a human labeler ranks all the responses

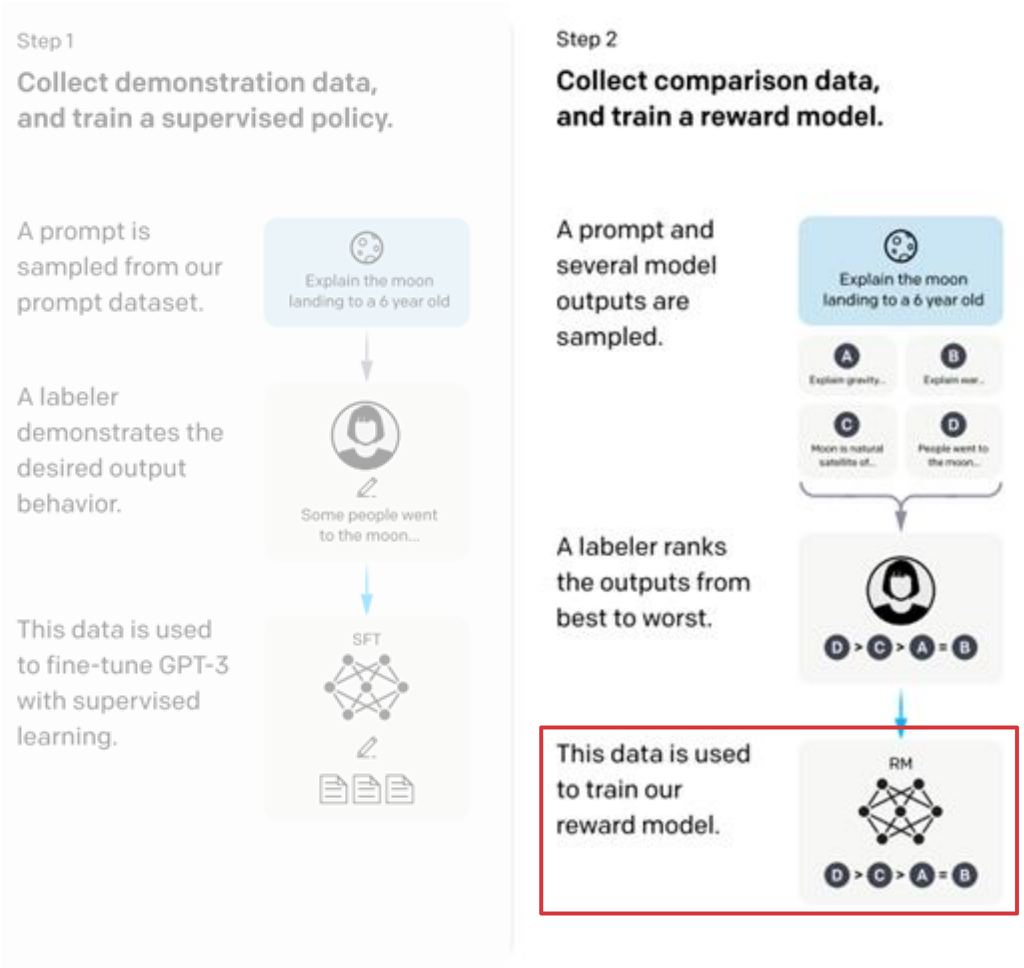
Source: https://arxiv.org/pdf/2203.02155

Reinforcement Learning from Human Feedback: Ranking Interface



Source: http://arxiv.org/abs/2204.05862

Reinforcement Learning from Human Feedback: Reward Model



- In Step 2, the model takes 33k prompts and samples $K \in \{4, \ldots, 9\}$ responses from the instruction fine-tuned model for each one
- Then, a human labeler ranks all the responses

Source: https://arxiv.org/pdf/2203.02155

Reinforcement Learning from Human Feedback: Reward Model

- The reward model is a smaller LLM (e.g., 6B vs. 175B parameters), but with the softmax over words replaced with a linear layer so that it outputs a scalar value, i.e. the reward of some (prompt, response) pair
- This *regression* model is trained so that rewards of higher-ranking responses, y_w, are larger than those of the lower-ranking responses, y_l:

$$\ell(\theta) = -\frac{1}{\binom{K}{2}} \mathbb{E}_{(x, y_w, y_l) \sim \mathcal{D}} [\log \sigma (r_\theta(x, y_w) - r_\theta(x, y_l))]$$

where x is the prompt, K is the number of responses for prompt x, $\mathcal{D}$ is the set of human labelled preferences, and r_θ is the reward model

Reinforcement Learning from Human Feedback: Reward Model

- The reward model is a smaller LLM (e.g., 6B vs. 175B parameters), but with the softmax over words replaced with a linear layer so that it outputs a scalar value, i.e. the reward of some (prompt, response) pair
- This *regression* model is trained so that rewards of higher-ranking responses, y_w, are larger than those of the lower-ranking responses, y_l:

$$\ell(\theta) = -\frac{1}{\binom{K}{2}} \mathbb{E}_{(x,y_w,y_l) \sim \mathcal{D}}[\log \sigma(r_\theta(x, y_w) - r_\theta(x, y_l))]$$

- During training, all $\binom{K}{2}$ pairwise rankings for a prompt are kept together in a batch for efficiency/stability

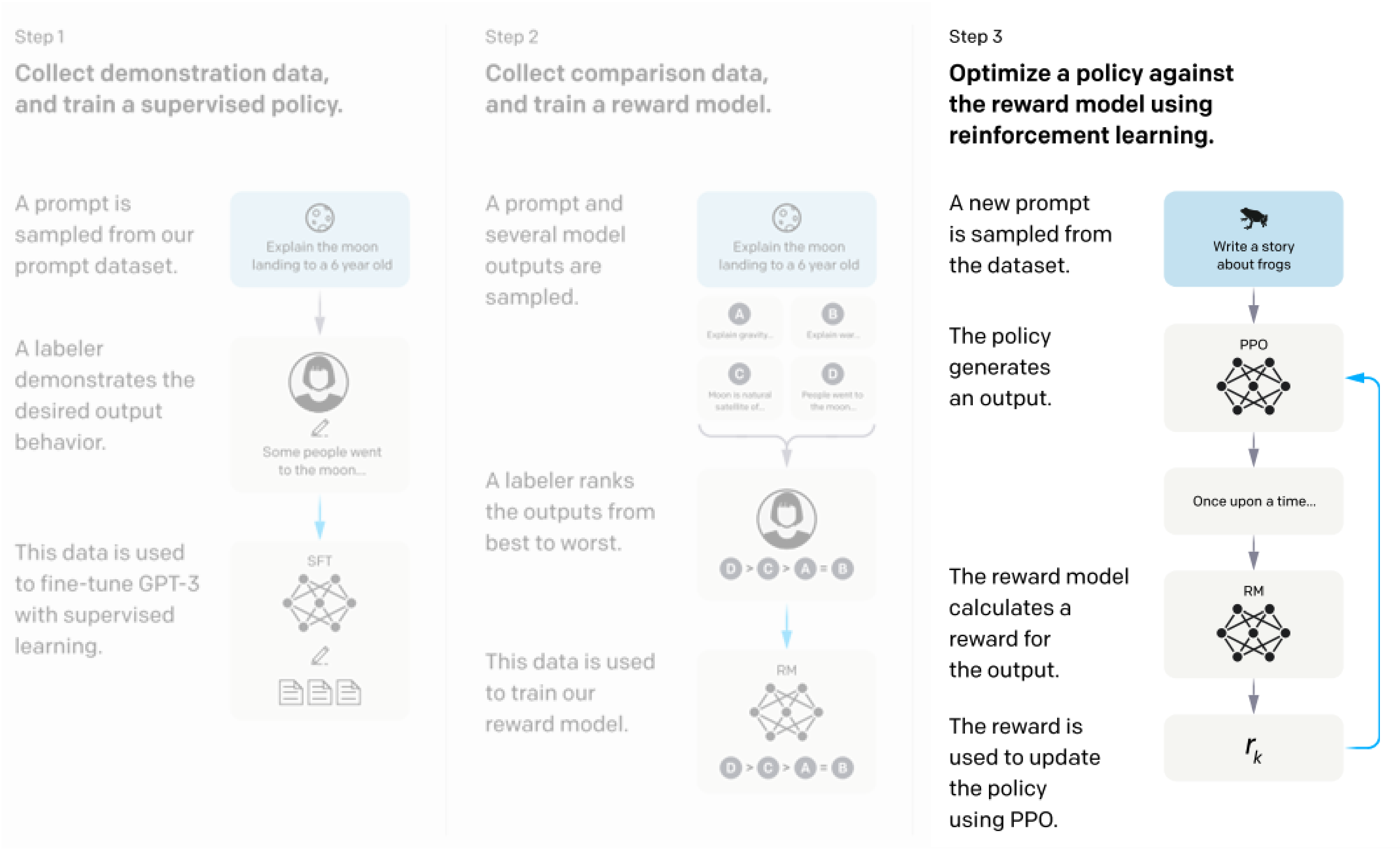
Okay so now what do we do with this thing…?

- The reward model is a smaller LLM (e.g., 6B vs. 175B parameters), but with the softmax over words replaced with a linear layer so that it outputs a scalar value, i.e. the reward of some (prompt, response) pair
- This *regression* model is trained so that rewards of higher-ranking responses, y_w, are larger than those of the lower-ranking responses, y_l:

$$\ell(\theta) = -\frac{1}{\binom{K}{2}} \mathbb{E}_{(x,y_w,y_l)\sim\mathcal{D}}[\log\sigma(r_\theta(x, y_w) - r_\theta(x, y_l))]$$

- During training, all $\binom{K}{2}$ pairwise rankings for a prompt are kept together in a batch for efficiency/stability

Okay so now what do we do with this thing…?



Source: https://arxiv.org/pdf/2203.02155

Reinforcement Learning: Problem Formulation

- State space, $\mathcal{S}$
- Action space, $\mathcal{A}$
- Reward function
  - Stochastic, $p(r \mid s, a)$
  - Deterministic, $R \colon \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$
- Transition function
  - Stochastic, $p(s' \mid s, a)$
  - Deterministic, $\delta \colon \mathcal{S} \times \mathcal{A} \rightarrow \mathcal{S}$

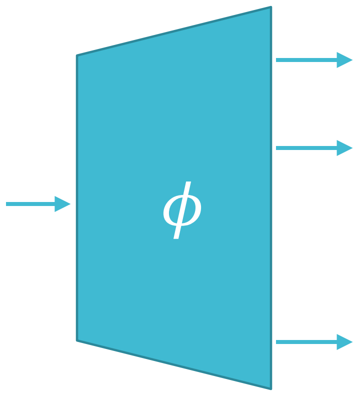
Reinforcement Learning: Problem Formulation for Fine-tuning LLMs

- State space, $\mathcal{S} = \{\text{all possible sequences of tokens}\}$
- Action space, $\mathcal{A} = \{\text{vocabulary of next tokens}\}$
- Reward function
  - Stochastic, $p(r \mid s, a)$
  - Deterministic reward based on reward model trained on human feedback, R_θ
    - R_θ is a bit of weird reward function from an RL perspective: it returns $0 \ \forall\ a \neq \text{EOS}$ and $r_\theta(x, [s, a] - x)$ otherwise
- Transition function
  - Stochastic, $p(s' \mid s, a)$
  - Deterministic, $\delta(s, a) = [s, a]$

Reinforcement Learning: Object of Interest

- A stochastic, parametrized policy, $\pi_\phi(a \mid s)$
  - Specifies a distribution over actions in *every* state

Model: s_t



$$p(a_1|s_t;\phi) \coloneqq \pi_\phi(a_1|s_t)$$
$$p(a_2|s_t;\phi) \coloneqq \pi_\phi(a_2|s_t)$$
$$\vdots$$
$$p(a_{|\mathcal{A}|}|s_t;\phi) \coloneqq \pi_\phi(a_{|\mathcal{A}|}|s_t)$$

- An *episode* $\mathrm{T} = \{s_0, a_0, s_1, a_1, \dots, s_T\}$ is one run of an agent through the system ending in a terminal state, $\boldsymbol{s_T}$

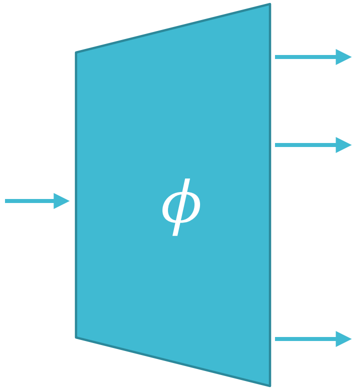
- The stochastic policy induces a distribution over episodes

$$p_\phi(\mathrm{T}) = p(\{s_0, a_0, s_1, a_1, \dots, s_T\})$$
$$= p(s_0) \prod_{t=0}^{T-1} p(s_{t+1} \mid s_t, a_t)\, \pi_\phi(a_t|s_t)$$

Reinforcement Learning: Object of Interest for Fine-tuning LLMs

- **The LLM to be fine-tuned,** $\pi_\phi(a \mid s)$
  - Specifies a distribution over next tokens given any input sequence

Model: s_t



$$p(a_1|s_t;\phi) \coloneqq \pi_\phi(a_1|s_t)$$
$$p(a_2|s_t;\phi) \coloneqq \pi_\phi(a_2|s_t)$$
$$\vdots$$
$$p(a_{|\mathcal{A}|}|s_t;\phi) \coloneqq \pi_\phi(a_{|\mathcal{A}|}|s_t)$$

- An *episode* $\mathrm{T} = \{x, a_0, s_1, a_1, \ldots, s_T\}$ is one completion of the prompt x, ending in an EOS token
- The LLM induces a distribution over possible completions

$$p_\phi(\mathrm{T}) = p(\{a_0, s_1, a_1, \ldots, s_T\} \mid x \coloneqq s_0)$$
$$= \prod_{t=0}^{T-1} \pi_\phi(a_t|s_t)$$

Objective function: $\ell(\phi) = -\mathbb{E}_{p_\phi(\mathrm{T})}[R_\theta(\mathrm{T})]$, the negative expected reward of a response

Policy Gradient Methods

$$\nabla_\phi \ell(\phi) = \nabla_\phi \left(-\mathbb{E}_{p_\phi(\mathrm{T})}[R_\theta(\mathrm{T})]\right) = \nabla_\phi \left(-\int R_\theta(\mathrm{T}) p_\phi(\mathrm{T})\, d\mathrm{T}\right)$$

$$= -\int R_\theta(\mathrm{T}) \nabla_\phi \left(\prod_{t=0}^{T-1} \pi_\phi(a_t|s_t)\right) d\mathrm{T}$$

- Issue: $\nabla_\phi p_\phi(\mathrm{T})$ involves taking the gradient of a (hideous) product

Objective function: $\ell(\phi) = -\mathbb{E}_{p_\phi(\mathrm{T})}[R_\theta(\mathrm{T})]$, the negative expected reward of a response

Likelihood Ratio Method a.k.a. REINFORCE (Williams, 1992)

$$\nabla_\phi \ell(\phi) = \nabla_\phi \left(-\mathbb{E}_{p_\phi(\mathrm{T})}[R_\theta(\mathrm{T})]\right) = \nabla_\phi \left(-\int R_\theta(\mathrm{T}) p_\phi(\mathrm{T})\, d\mathrm{T}\right)$$

$$= -\int R_\theta(\mathrm{T}) \nabla_\phi \left(\prod_{t=0}^{T-1} \pi_\phi(a_t|s_t)\right) d\mathrm{T}$$

- Insight:

$$\nabla_\phi p_\phi(\mathrm{T}) = \frac{p_\phi(\mathrm{T})}{p_\phi(\mathrm{T})} \nabla_\phi p_\phi(\mathrm{T}) = p_\phi(\mathrm{T}) \nabla_\phi (\log p_\phi(\mathrm{T}))$$

$$\log p_\phi(\mathrm{T}) = \sum_{t=0}^{T-1} \log \pi_\phi(a_t|s_t)$$

$$\nabla_\phi (\log p_\phi(\mathrm{T})) = \sum_{t=0}^{T-1} \nabla_\phi \log \pi_\phi(a_t \mid s_t)$$

Source: https://link.springer.com/article/10.1007/bf00992696

Objective function: $\ell(\phi) = -\mathbb{E}_{p_\phi(\mathrm{T})}[R_\theta(\mathrm{T})]$, the negative expected reward of a response

Likelihood Ratio Method a.k.a. REINFORCE (Williams, 1992)

$$\nabla_\phi \ell(\phi) = \nabla_\phi \left(-\mathbb{E}_{p_\phi(\mathrm{T})}[R_\theta(\mathrm{T})]\right) = \nabla_\phi \left(-\int R_\theta(\mathrm{T}) p_\phi(\mathrm{T})\, d\mathrm{T}\right)$$

$$= -\int R_\theta(\mathrm{T}) \nabla_\phi p_\phi(\mathrm{T}) d\mathrm{T} = -\int R_\theta(\mathrm{T}) \nabla_\phi \big(\log p_\phi(\mathrm{T})\big) p_\phi(\mathrm{T}) d\mathrm{T}$$

$$= -\mathbb{E}_{p_\phi(\mathrm{T})}\left[R_\theta(\mathrm{T}) \nabla_\phi \big(\log p_\phi(\mathrm{T})\big)\right]$$

$$\approx -\frac{1}{N}\sum_{n=1}^{N} R_\theta\big(\mathrm{T}^{(n)}\big)\, \nabla_\phi\big(\log p_\phi\big(\mathrm{T}^{(n)}\big)\big)$$

(where $\mathrm{T}^{(n)} = \left\{a_0^{(n)}, s_1^{(n)}, a_1^{(n)}, \ldots, s_{T^{(n)}}^{(n)}\right\}$ is a sampled completion of x)

$$= -\frac{1}{N}\sum_{n=1}^{N} r_\theta\left(x, \left[a_0^{(n)}, \ldots, a_{T^{(n)}}^{(n)}\right]\right) \left(\sum_{t=0}^{T^{(n)}-1} \nabla_\phi \log \pi_\phi\left(a_t^{(n)} \middle| s_t^{(n)}\right)\right)$$