



10-423/10-623 Generative AI

Machine Learning Department
School of Computer Science
Carnegie Mellon University

Instruction Fine-tuning + Reinforcement Learning with Human Feedback (RLHF)

Pat Virtue & Matt Gormley

Lecture 11

Feb. 19, 2025

Few-shot Learning with LLMs

Suppose you have...

- a dataset $D = \{(x_i, y_i)\}_{i=1}^N$ and N is rather small (i.e. few-shot setting)
- a very large (billions of parameters) pre-trained language model

There are two ways to “learn”

Option A: Supervised fine-tuning (SFT)

Improve pre-trained LLM

Option B: In-context learning

Fixed pre-trained LLM

Last time

- Parameter efficient fine-tuning (PEFT)
 - How to fine-tune (efficiently)

Today

- Instruction fine-tuning (IFT)
 - What to fine tune on (instruction datasets)
- Reinforcement learning with human feedback (RLHF)

Today

- Prompt Engineering
 - Chain of thought prompting

PROMPT ENGINEERING

Prompt Engineering

- **Task:** News topic classification
- **Dataset:** AG News
- **Model:** OPT-175B
- **Setup:** zero-shot learning

Question: if we evaluate the model multiple times keeping everything fixed except for the prompt, do we always get the same results?

| 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 |

Prompt Engineering

- **Task:** News topic classification
- **Dataset:** AG News
- **Model:** OPT-175B
- **Setup:** zero-shot learning

Question: how can we pick a good prompt?

Answer: pick the prompt with the lowest perplexity under the model!

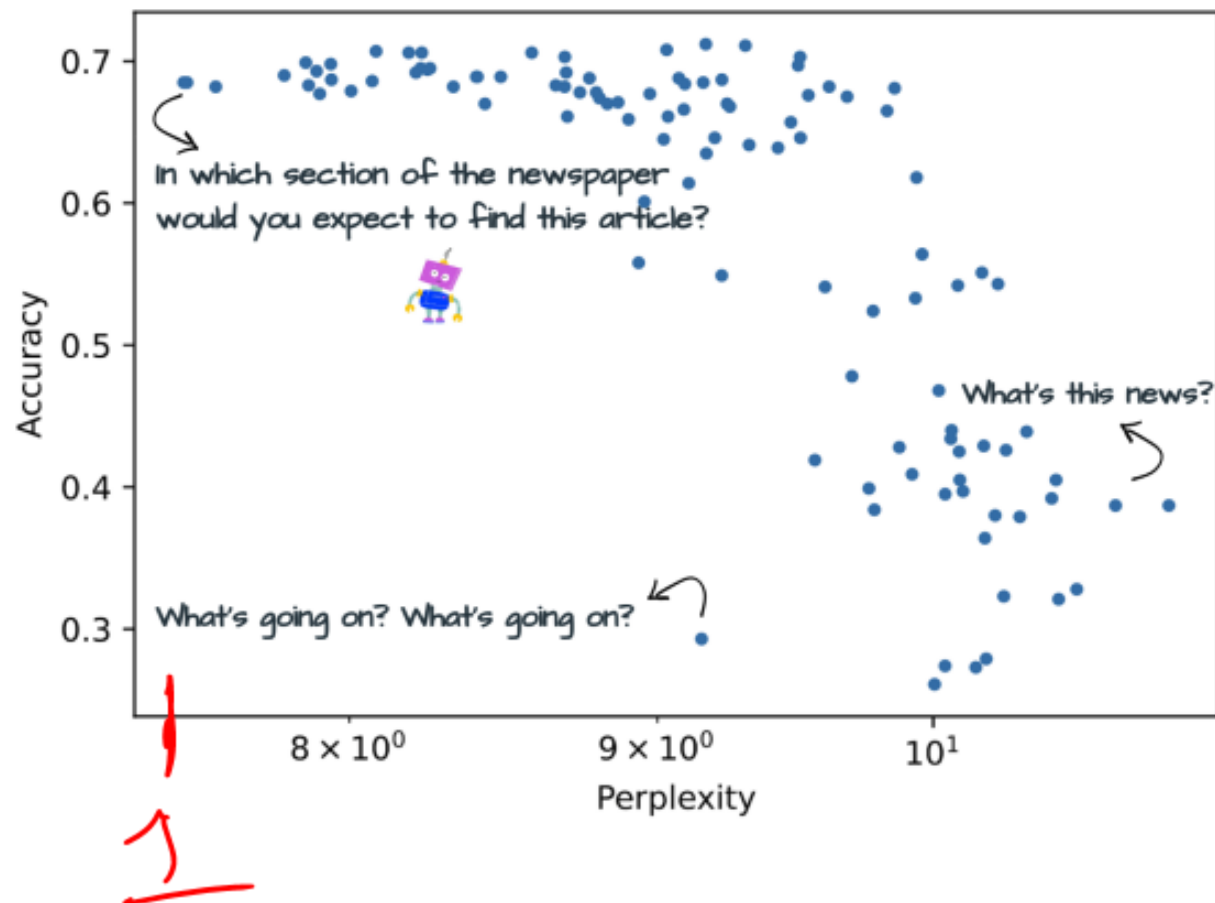


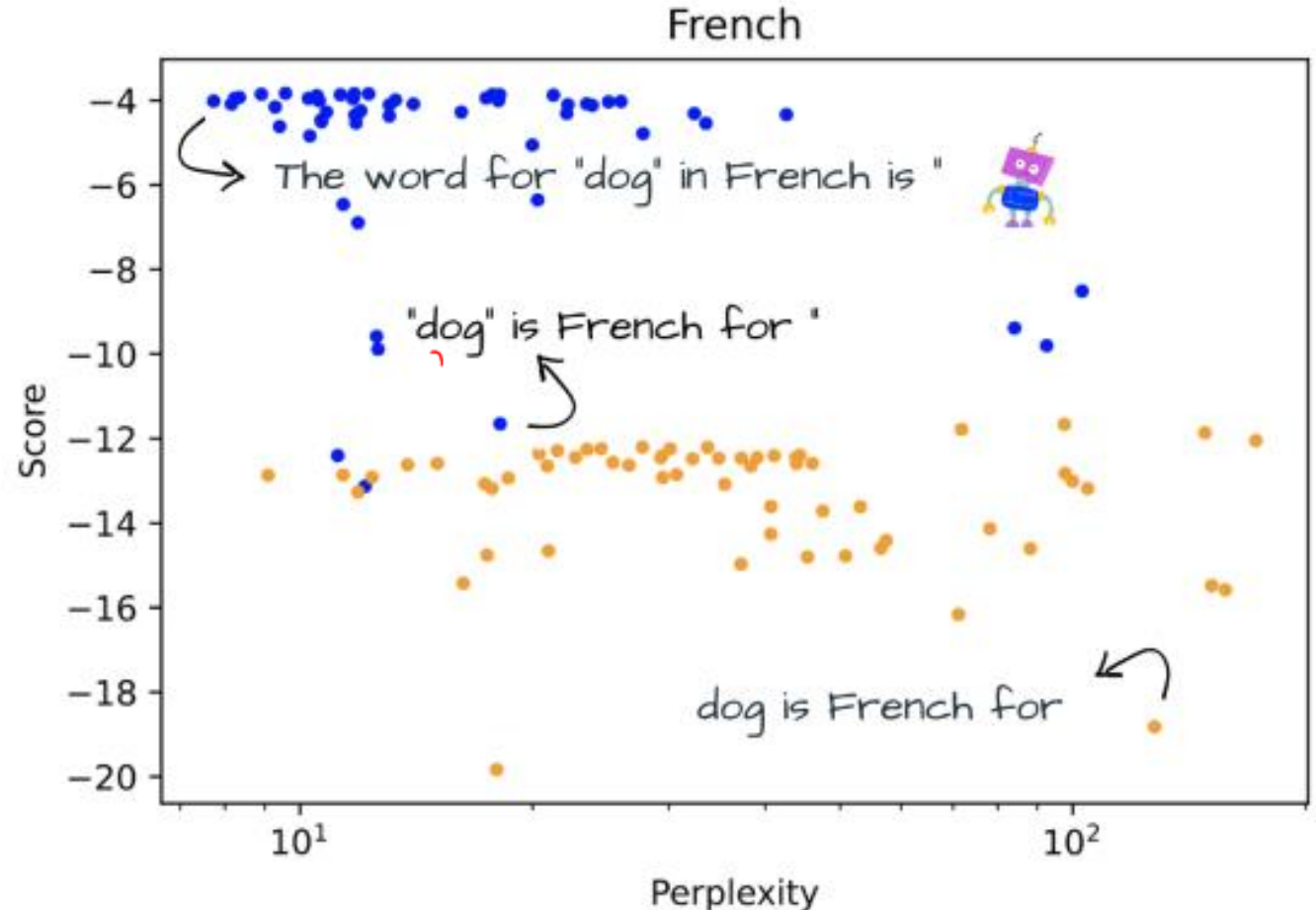
Figure 1: Accuracy vs. perplexity for the AG News dataset with OPT 175b. The x axis is in log scale. Each point stands for a different prompt.

Prompt Engineering

- **Task:** French word-level translation
- **Dataset:** NorthEuraLex
- **Model:** Bloom (multilingual LLM)
- **Setup:** zero-shot learning

Question: how can we pick a good prompt?

Answer: pick the prompt with the lowest perplexity under the model!



CHAIN-OF-THOUGHT PROMPTING

Chain-of-Thought Prompting

Standard Prompting

Model Input

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

A: The answer is 11.

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

Model Output

A: The answer is 27. ❌

Chain-of-Thought Prompting

Model Input

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

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

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

Model Output

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

Train

Test

Chain-of-Thought Prompting

- Asking the model to reason about its answer can improve its performance for few-shot in-context learning
- **Chain-of-thought prompting** provides such reasoning in the in-context examples

| Standard Prompting | Chain-of-Thought Prompting |
|---|---|
| <p>Model Input</p> <p>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?</p> <p>A: The answer is 11.</p> <p>Q: The cafeteria had 23 apples. If they used 20 to make lunch and bought 6 more, how many apples do they have?</p> | <p>Model Input</p> <p>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?</p> <p>A: Roger started with 5 balls. 2 cans of 3 tennis balls each is 6 tennis balls. $5 + 6 = 11$. The answer is 11.</p> <p>Q: The cafeteria had 23 apples. If they used 20 to make lunch and bought 6 more, how many apples do they have?</p> |
| <p>Model Output</p> <p>A: The answer is 27. ❌</p> | <p>Model Output</p> <p>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. ✅</p> |

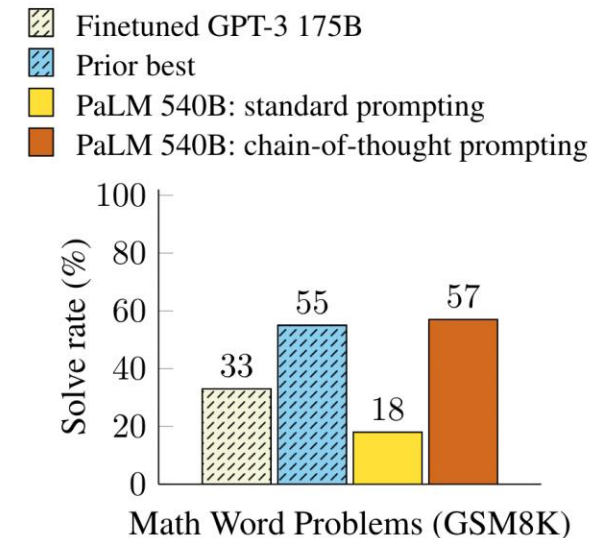


Figure 2: PaLM 540B uses chain-of-thought prompting to achieve new state-of-the-art performance on the GSM8K benchmark of math word problems. Finetuned GPT-3 and prior best are from Cobbe et al. (2021).

Chain-of-Thought Prompting

- Asking the model to reason about its answer can improve its performance for few-shot in-context learning
- **Chain-of-thought prompting** provides such reasoning in the in-context 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.* ❌

(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* ❌

(d) Zero-shot-CoT (Ours)

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.**

(Output) *There are 16 balls in total. Half of the balls are golf balls. That means that there are 8 golf balls. Half of the golf balls are blue. That means that there are 4 blue golf balls.* ✔️

- But the model does better even if you just prompt it to reason step-by-step

Chain-of-Thought Prompting

(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**

(d) Zero-shot-CoT (Ours)

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.**

(Output) There are 16 balls in total. Half of the balls are golf balls. That means that there are 8 golf balls. Half of the golf balls are blue. That means that there are 4 blue golf balls. **✓**

Chain-of-Thought Prompting

- Asking the model to reason about its answer can improve its performance for few-shot in-context learning
- **Chain-of-thought prompting** provides such reasoning in the in-context examples

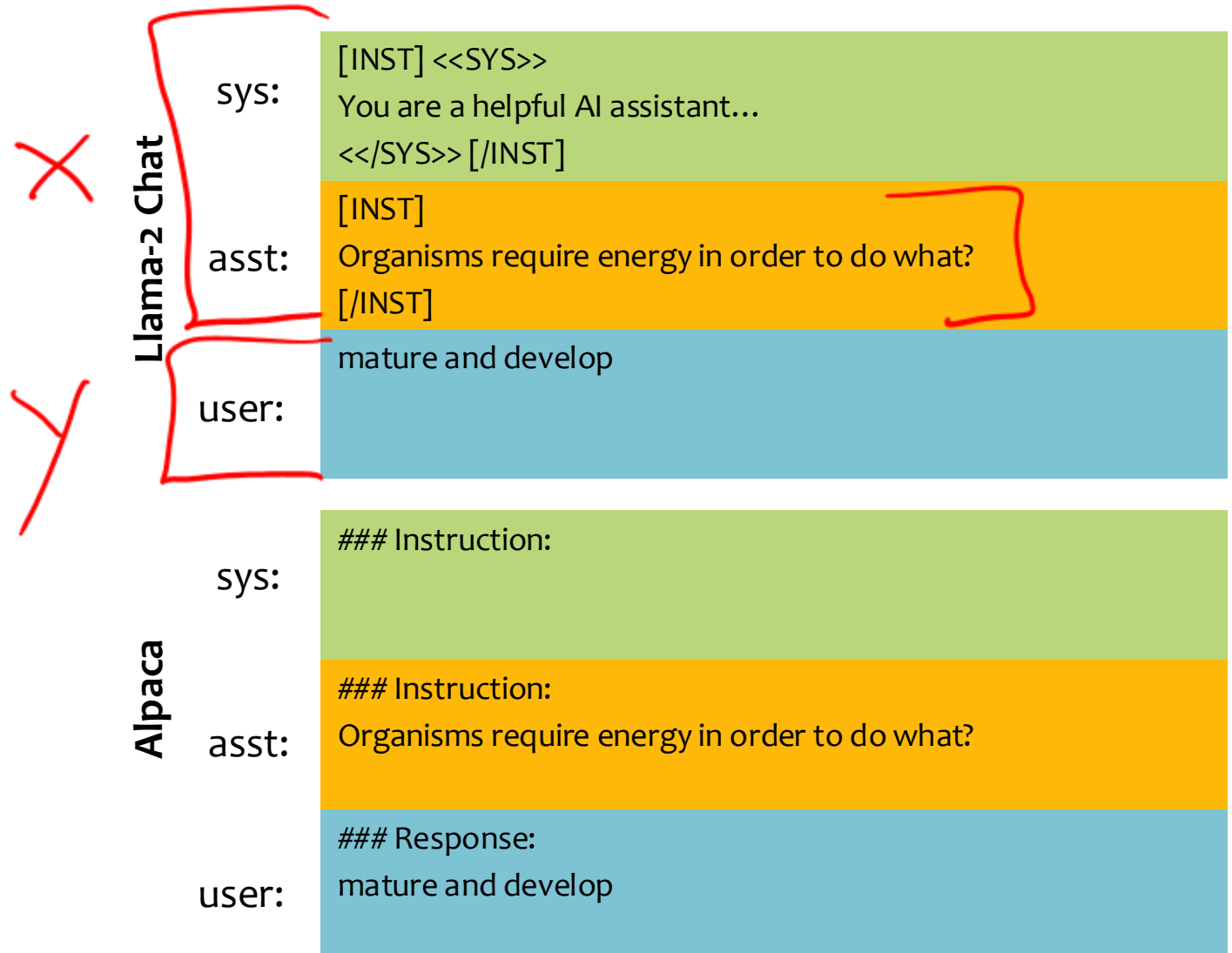
| | MultiArith | GSM8K |
|--------------------------|-------------|-------------|
| Zero-Shot | 17.7 | 10.4 |
| Few-Shot (2 samples) | 33.7 | 15.6 |
| Few-Shot (8 samples) | 33.8 | 15.6 |
| Zero-Shot-CoT | 78.7 | 40.7 |
| Few-Shot-CoT (2 samples) | 84.8 | 41.3 |

- But the model does better even if you just prompt it to reason step-by-step

INSTRUCTION FINE-TUNING

Prompting for Instruction Fine-tuned Models

- Models like ChatGPT, Llama-2 Chat, etc. have been fine-tuned as chat assistants
- These (often) were trained with specific prompt templates that segment the prompt into different parts: (1) system (2) assistant (3) user



Prompting for Instruction Fine-tuned Models

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:

Llama-2-70B

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

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.

Model Output

Prompting for Instruction Fine-tuned Models

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 necessary items, including swimsuits, towels, and sunscreen. Jason and the kids went to the beach, driving there. The truck arrived, and they spent a fun-filled day at the beach, playing and splashing in the surf. One-sentence summary:

Llama-2-7B Chat was *instruction fine-tuned* (more on this later) and so its responses look quite different from the Llama-2 models that were not



Model Output

Llama-2-70B

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

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.

Instruction Fine-Tuning

- Motivation: Autocomplete → e.g. Chat
 - Suppose you want to build a chat agent
 - LLMs are trained to reduce the perplexity of a large training corpus containing web text, articles, code, etc. (i.e. it's good at completing your _____)
 - But a chat agent should not merely predict what comes next, it should behave conversationally and know when to stop
 - We want to align the LLM with the expectations of a human user for a given task in some *instruction*
- Key idea:
 - Build a “chat agent” training dataset
 - Fine-tune the LLM on this data
- This technique goes by many names...
 - instruction fine-tuning
 - chat fine-tuning
 - alignment
 - behavioral fine-tuning

SFT

Instruction Fine-Tuning

Question:

How can we build a “chat agent” training dataset?

Answer:

Sources of prompts:

Humans / web conv
LLM (bigger/smaller)

Sources of responses:

Humans / web conv
LLM

- Key idea:
 - Build a “chat agent” training dataset
 - Fine-tune the LLM on this data
- This technique goes by many names...
 - instruction fine-tuning
 - chat fine-tuning
 - alignment
 - behavioral fine-tuning

Datasets for Instruction Fine-Tuning

| Release | Collection | Model Details | | | | Data Collection & Training Details | | | |
|---------|---------------------|--------------------|-------------|----------|---------|------------------------------------|---------------|-------|---|
| | | Model | Base | Size | Public? | Prompt Types | Tasks in Flan | # Exs | Methods |
| 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 | MetalCL | MetalCL | 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.

Dataset: InstructGPT

- InstructGPT consisted of 13k prompt/response pairs
- Labelers were tasked with writing an instruction prompt and a demonstration response of how a chat agent should reply
- 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% |


Dataset: Dolly

- Dolly is an open source follow-up to InstructGPT of 15k instruction fine-tuning examples
- 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.

Dataset: Dolly

| Category | Instruction | Context | 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. |

Dataset: Dolly

| Category | Instruction | Context | Response |
|-------------|---|---------|--|
| General Q&A | What is <u>best</u> 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. |



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. |

Dataset: Dolly

| Category | Instruction | Context | Response |
|------------------------|--|--|---|
| Information Extraction | <u>Extract</u> 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. | <ul style="list-style-type: none">• November 16, 1813 - This is when Daniel Leavitt was born• 1838 - This is when Daniel Leavitt married his wife Ruth Jeannette Ball. |

Dataset: Dolly

| Category | Instruction | Context | 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. |

Dataset: Dolly

| Category | Instruction | Context | Response |
|---------------|-----------------------------|--|--|
| Summarization | What is a dispersive prism? | <p>In optics, a dispersive prism is an optical prism that is used to disperse light, that is, to separate light into its spectral components (the colors of the rainbow). Different wavelengths (colors) of light will be deflected by the prism at different angles. This is a result of the prism material's index of refraction varying with wavelength ...</p> | <p>A dispersive prism is an optical prism that disperses the light's different wavelengths at different angles. When white light is shined through a dispersive prism it will separate into the different colors of the rainbow.</p> |

Dataset: Dolly

| Category | Instruction | Context | Response |
|----------------|--|---------|---|
| Classification | Identify which instrument is string or percussion: Cantaro, Gudok | | Gudok is string, Cantaro is percussion. |

Dataset: Dolly

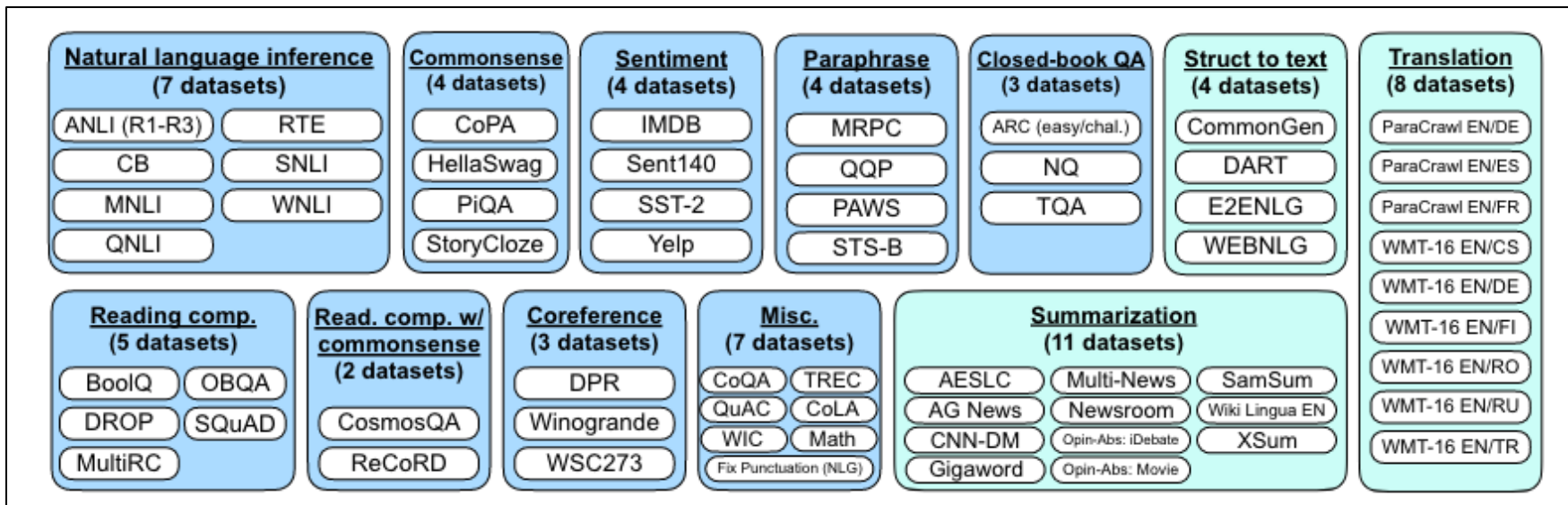
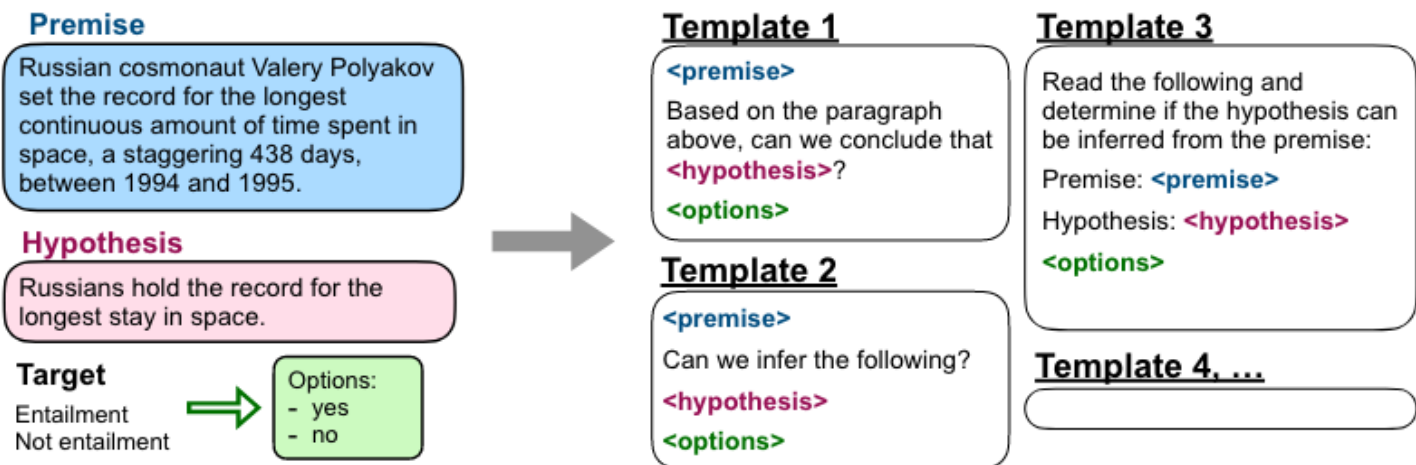
| Category | Instruction | Context | Response |
|------------------|--|---------|---|
| Creative Writing | Write a haiku about how much I love pickles. | | Caressed by sweet brine, you are crispy and crunchy. Oh, won't you be mine? |

Dataset: Dolly

| Category | Instruction | Context | Response |
|------------------|--|---------|---|
| 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. |

- Flan (Wei et al., 2021) was one of the first instruction fine-tuning datasets
- Recent versions of the dataset contain ~3.5 million examples
- **Key idea:** build instruction fine-tuning examples from existing NLP tasks/datasets (12 tasks, 62 datasets)
- For each NLP task/dataset, Flan created 10 different templates

Dataset: Flan



Instruction Fine-Tuned Models

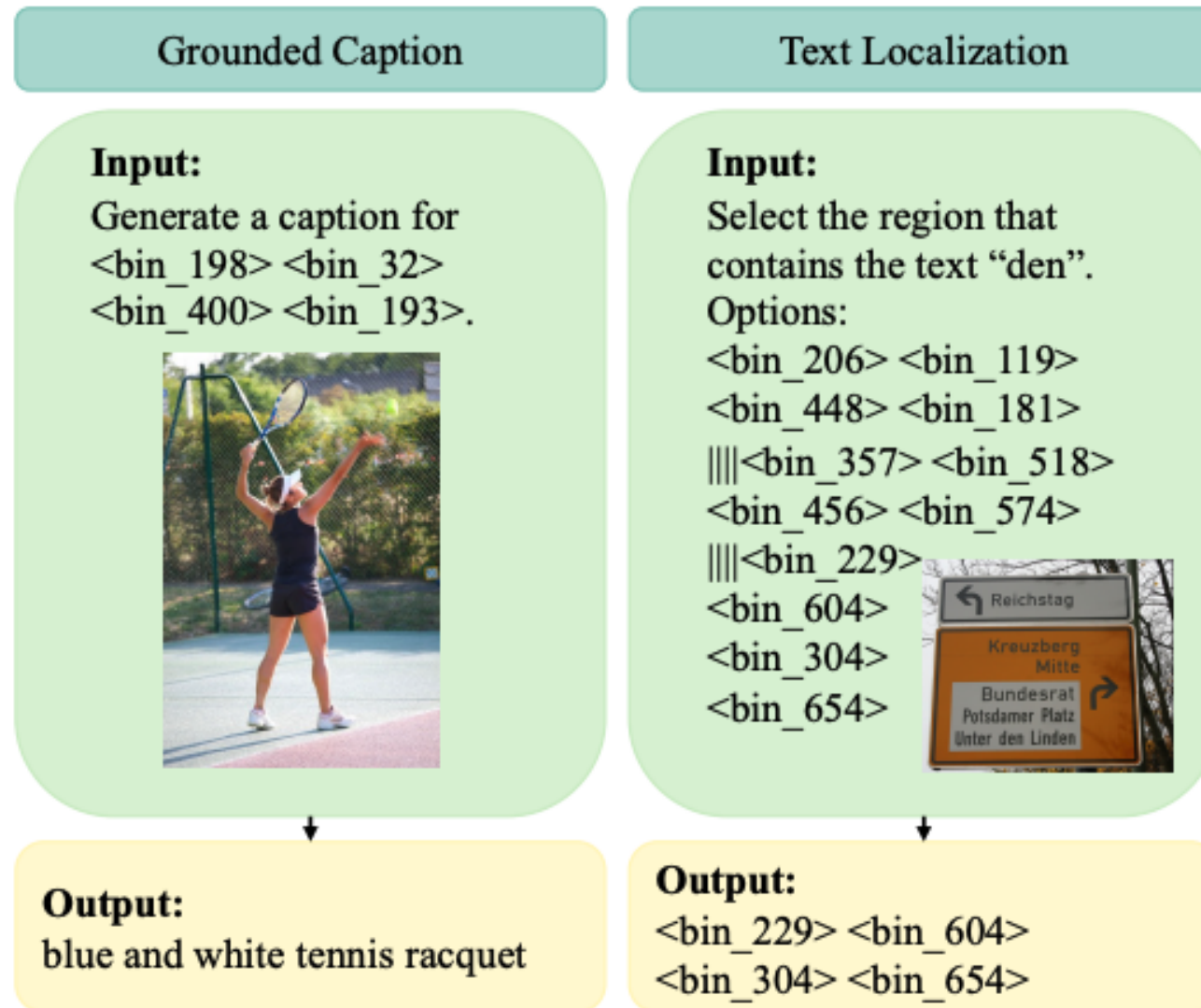
These models begin with a pre-trained Base Model

They are then fine-tuned on some instruction following dataset

The resulting model is often very effective even at a smaller scale than the largest LLMs available (e.g. 7B – 13B parameters is typical)

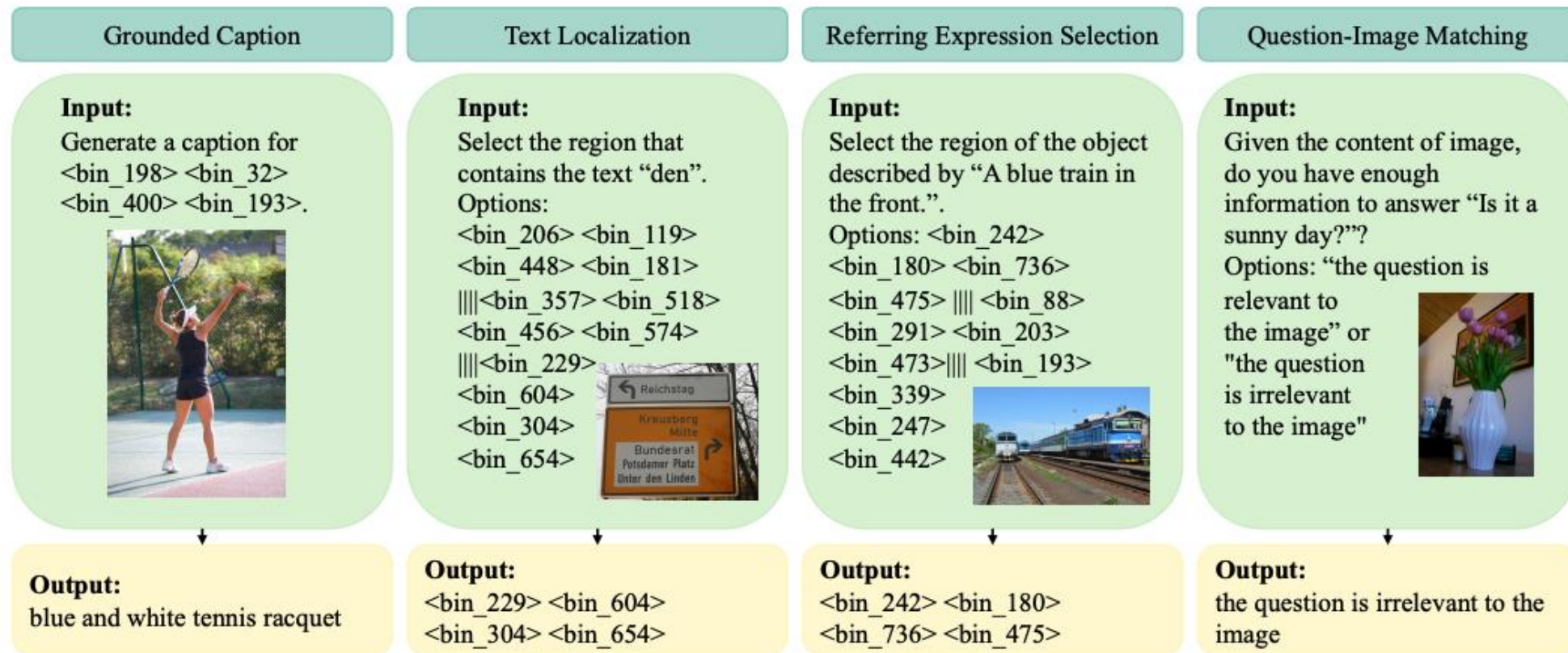
| Instruction fine-tuned LLMs | # Params | Base Model | Fine-tuning Trainset | | |
|--|----------|-----------------------------------|----------------------|---------------|------------|
| | | | Self-build | Dataset Name | Size |
| Instruct-GPT (Ouyang et al., 2022) | 176B | GPT-3 (Brown et al., 2020b) | 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 | - | - |

Multi-Modal Instruction Fine-Tuning



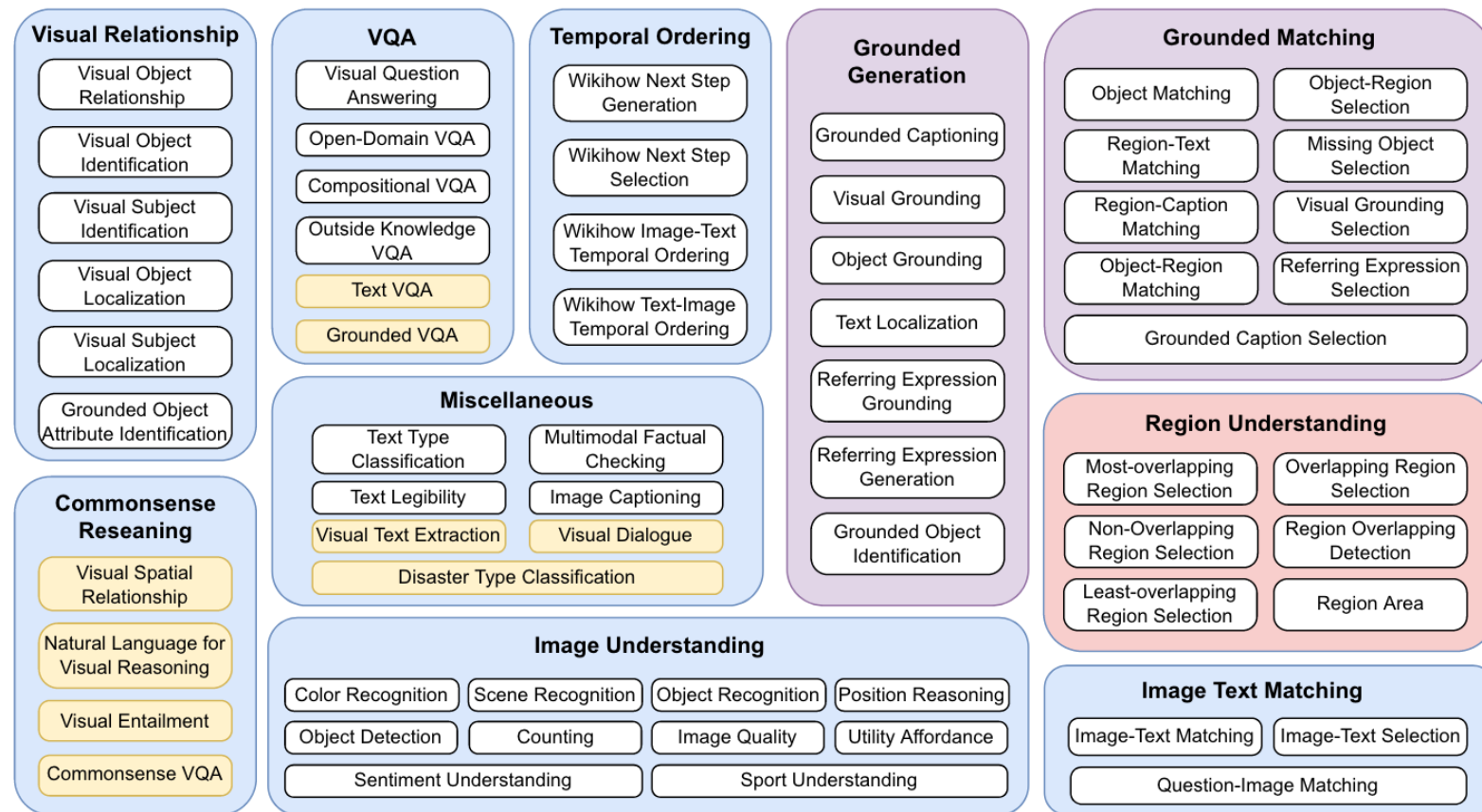
Multi-Modal Instruction Fine-Tuning

MultilInstruct (Xu et al., 2023) combines 62 multi-modal tasks from 21 open source datasets into a single multi-modal instruction fine-tuning dataset



Multi-Modal Instruction Fine-Tuning

MultilInstruct (Xu et al., 2023) combines 62 multi-modal tasks from 21 open source datasets into a single multi-modal instruction fine-tuning dataset



REINFORCEMENT LEARNING WITH HUMAN FEEDBACK (RLHF)

RLHF



RL

- **InstructGPT** uses Reinforcement Learning with Human Feedback (RLHF) to **fine-tune** a **pre-trained** GPT model
- From the paper: “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.”

Step 1
Collect demonstration data, and train a supervised policy.

A prompt is sampled from our prompt dataset.

Explain the moon landing to a 6 year old

A labeler demonstrates the desired output behavior.

Some people went to the moon...

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

SFT

IFT
SFT

Step 2
Collect comparison data, and train a reward model.

A prompt and several model outputs are sampled.

Explain the moon landing to a 6 year old

A Explain gravity... B Explain war...
C Moon is natural satellite of... D People went to the moon...

A labeler ranks the outputs from best to worst.

D > C > A = B

This data is used to train our reward model.

RM
D > C > A = B

$R(s, a)$

Step 3 RL
Optimize a policy against the reward model using reinforcement learning.

A new prompt is sampled from the dataset.

Write a story about frogs

The policy generates an output.

PPO

Once upon a time...

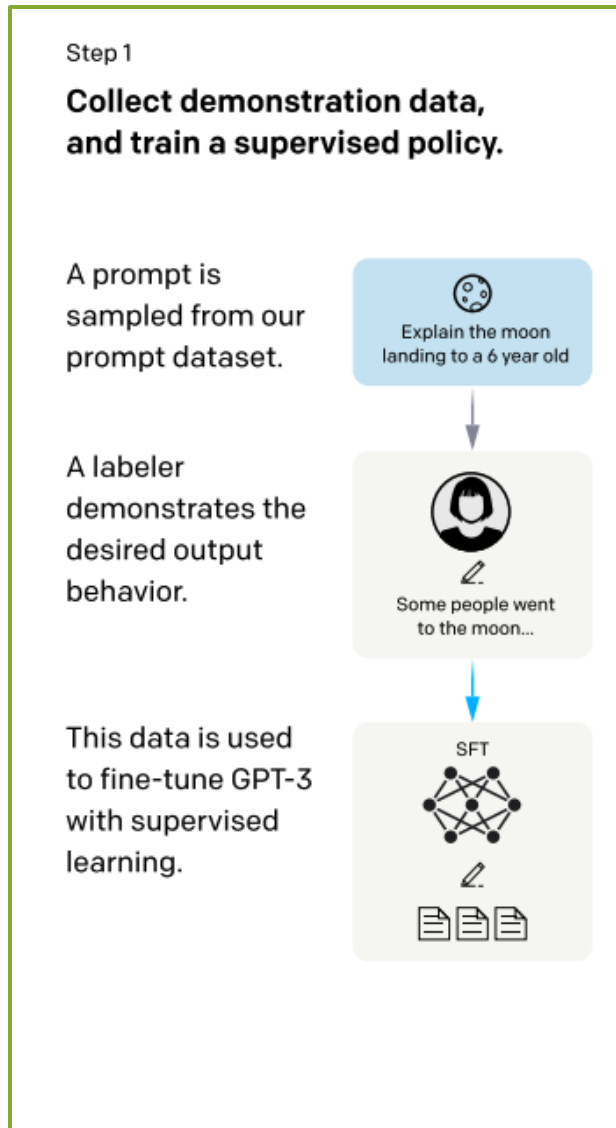
The reward model calculates a reward for the output.

RM

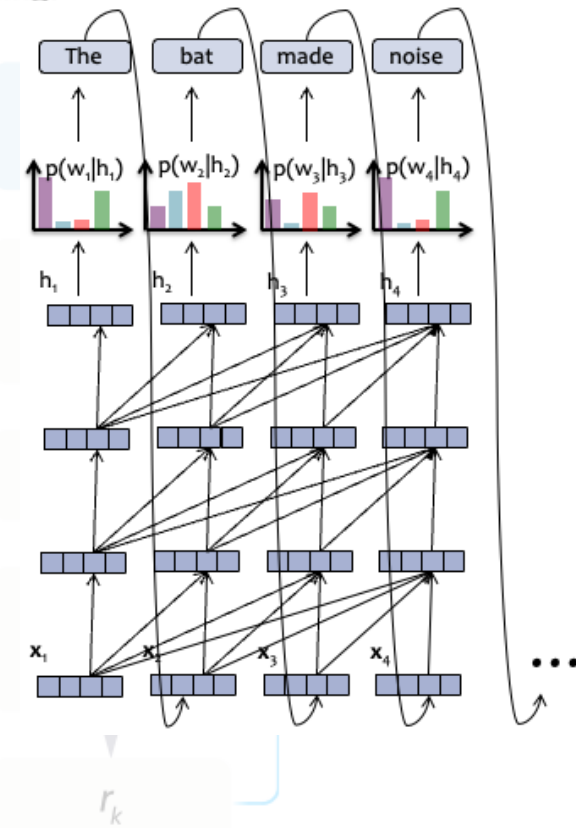
The reward is used to update the policy using PPO.

r_k

RLHF



- 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 interactions might still be limited by the contents of the training data



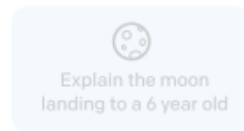
using PPO.

RLHF

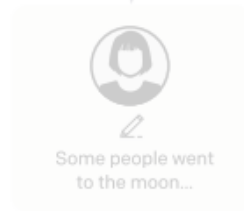
Step 1

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

A prompt is
sampled from our
prompt dataset.



A labeler
demonstrates the
desired output
behavior.



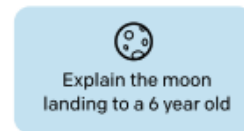
This data is used
to fine-tune GPT-3
with supervised
learning.



Step 2

**Collect comparison data,
and train a reward model.**

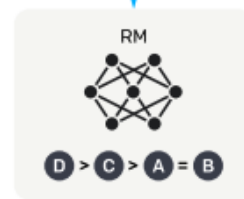
A prompt and
several model
outputs are
sampled.



A labeler ranks
the outputs from
best to worst.



This data is used
to train our
reward model.



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

Playground task ⓘ

Talk to the assistant

Task 1 of 5

Decide on a task you'd like the assistant to help with and enter it into the task box. Interact with the AI assistant. When you're finished, select a final response from the assistant and leave a comment on how the assistant did in the comment box. Click to go to the next task.

Comments

I thought the assistant was ...

Rating

Bad Good

Next Task

- Human**
I have to come up with a conversation with an AI assistant for the purposes of showing it off in a demo. What sorts of conversation topics would you recommend?
- Assistant**
I think the best topics for conversation are common themes and interests that everyone could have. Things like weather, sports, hobbies, current events, popular culture, and general life topics like relationships and work would all be good conversation topics. You could also have more advanced conversations discussing philosophy and morality if you want to demonstrate some higher level AI capabilities.
- Human**
I think I'm more interested in an advanced conversation. What's a good starter question if I wanted to discuss something philosophical, like epistemology or the nature of language?
- Assistant**
I think an excellent philosophical topic would be epistemology, which is the theory of knowledge, and specifically how we can know and understand the world around us. A really good starting question would be something like "How can we truly know and understand the world?", or "What is the nature of knowledge?". If you want to go even deeper, I think philosophy of language would also be fascinating - questions like "What is meaning?", "How do language and thought relate?", and "How can language mislead and deceive?" would be great discussion topics.
- Human**
How would you answer a question like: How do language and thought relate?

Choose the most helpful and honest response

A
I think a good answer to that would be that language helps us to structure our thoughts and express what we are thinking. Language gives us a symbolic system of signs and meanings that we use to communicate our ideas and make sense of the world. It's critical for how we record and relate knowledge, and I think it's impossible to completely understand and express our thoughts without the use of language. Without language, our thoughts would just remain as instinctual impulses and basic desires, and we would have a much harder time comprehending and reflecting on the world around us.

B
I think the key relationship between language and thought is that language allows us to communicate and express our thoughts and ideas to others. Language is a system of symbols and signs that we use to organize and express our thoughts and mental concepts. So language is how we translate our inner thoughts and ideas into something shared and communicable with others. Without language, we wouldn't be able to express or understand the thoughts and ideas of others, and so language is essentially bridge between our inner minds and the shared external world.

A A A A B B B B

A is better B is better

A ← + + → B

Example Interface for Crowdworkers

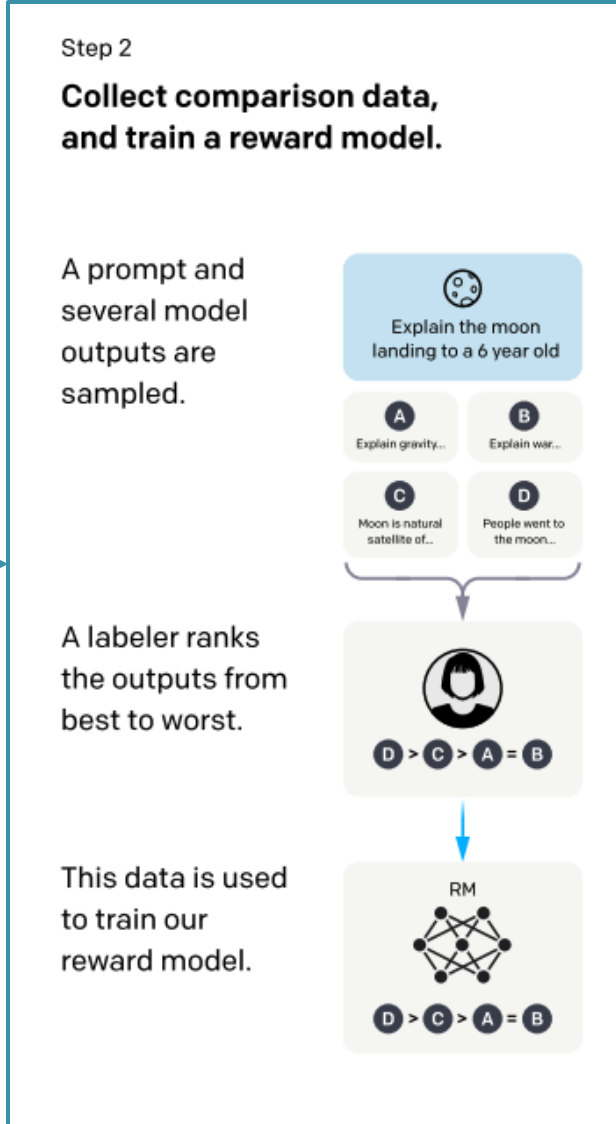
A

B

RLHF

$$r_{\theta}(x, y)$$

- The reward model is a copy of the Step-1 LLM, but with the softmax over words replaced so that it outputs a single scalar value, i.e. the reward
- The model is trained so that rewards of the higher ranking (winning) responses are larger than those of the lower ranking (losing) responses



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

RLHF

$r_{\theta}(x, y)$ ←

- The reward model is a copy of the Step-1 LLM, but with the softmax over words replaced so that it outputs a single scalar value, i.e. the reward
- This regression model is trained so that rewards of the higher ranking (winning) responses are larger than those of the lower ranking (losing) responses

- The objective function for the reward model:

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

- where

- x is the prompt

- y_w, y_l are the responses

- w denotes the winner, l the loser

- $r_{\theta}(x, y_l)$ is the output of the reward model

- D is the dataset of human rankings

- all the $(K \text{ choose } 2)$ rankings for each prompt are kept together in a single batch for efficiency/stability

win

loser



RLHF

- Step 3 trains the model from Step 1 using reinforcement learning
- Instead of having a human or some expert model provide rewards, we take the reward model from Step 2 as "ground truth" for the rewards
- Reinforcement learning uses (state, action, reward) tuples as training data
 - state = prompt
 - action = response
 - reward = scalar from regression reward model
 - each episode lasts exactly one turn
- RL objective is combined with pre-training objective:

$$\text{objective}(\phi) = \mathbb{E}_{(x,y) \sim D_{\pi_{\phi}^{RL}}} \left[r_{\theta}(x,y) - \beta \log \left(\frac{\pi_{\phi}^{RL}(y|x)}{\pi_{\phi}^{SFT}(y|x)} \right) \right] + \gamma \mathbb{E}_{x \sim D_{\text{pretrain}}} [\log(\pi_{\phi}^{RL}(x))]$$

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.



Once upon a time...


The reward model calculates a reward for the output.



The reward is used to update the policy using PPO.



RLHF Objective Function

$$\text{objective}(\phi) = \mathbb{E}_{(x,y) \sim D_{\pi_{\phi}^{RL}}} \left[r_{\theta}(x,y) - \beta \log \left(\frac{\pi_{\phi}^{RL}(y|x)}{\pi_{\phi}^{SFT}(y|x)} \right) \right] \\ + \gamma \mathbb{E}_{x \sim D_{\text{pretrain}}} [\log(\pi_{\phi}^{RL}(x))]$$


The objective function used here is modeled off of the (rather popular) [PPO algorithm](#). That algorithm, in turn, is a type of policy gradient method and motivated by the objective functions for [trust region policy optimization \(TRPO\)](#). But the (super high level) intuition behind the objective function is as follows:

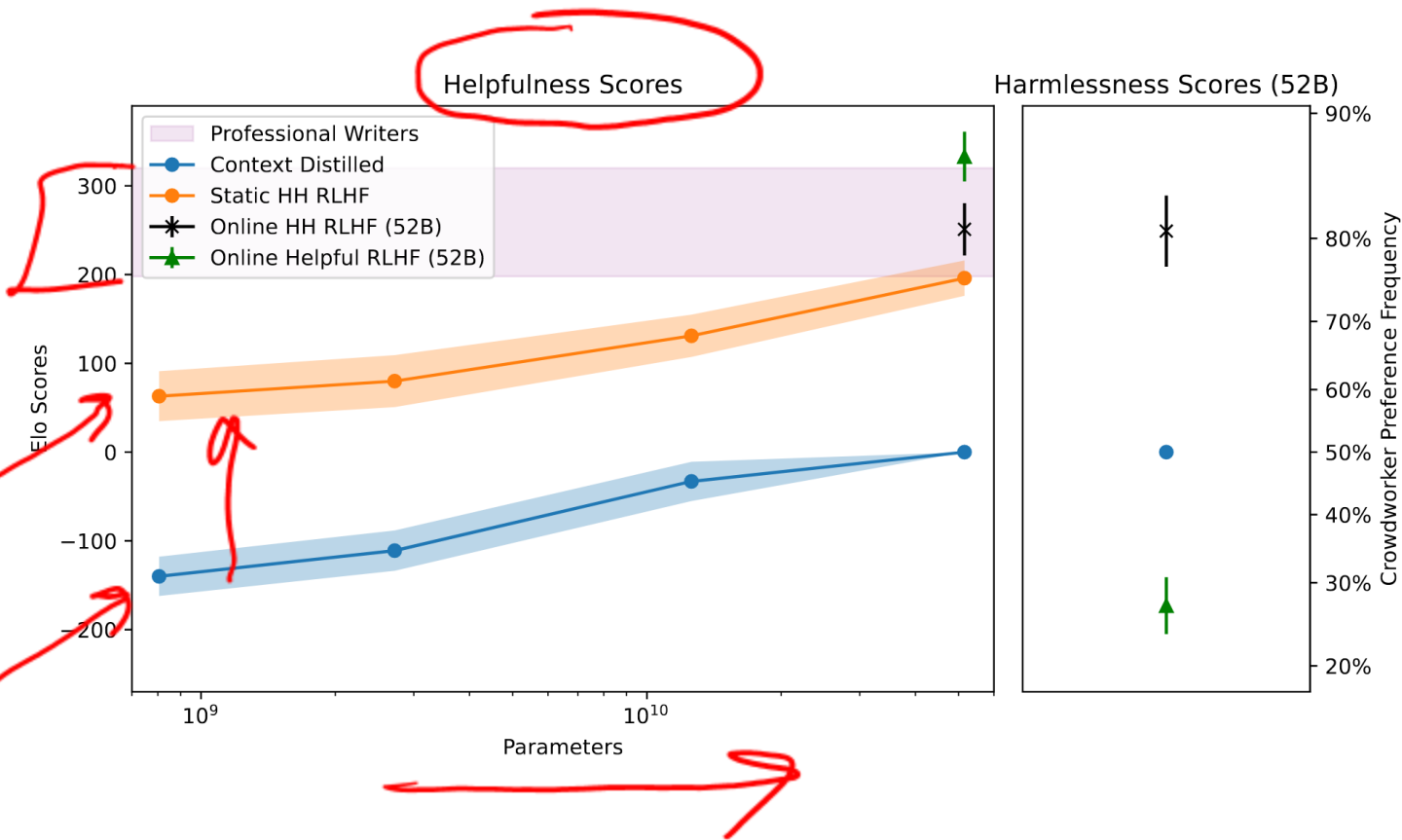
1. The expectation of the reward says that on samples from the RL trained model π_{RL} , we want the probability of that sample π_{RL} to be high when the reward r_{θ} is high and for it to be low otherwise.
2. The expectation of the beta term says that we don't want the RL trained model probabilities π_{RL} to stray to far from the supervised fine-tuned (SFT) model π_{SFT} -- this is instantiated as a KL divergence penalty.
3. The expectation under the pretraining distribution D_{pretrain} is just the standard log-likelihood of a training sample that we use for supervised fine-tuning, but applied here to the RL trained model as well.

Note that in practice, we don't compute these expectations exactly, we approximate each with a Monte Carlo approximation (i.e. a sum over a very small number of samples).

RLHF Results

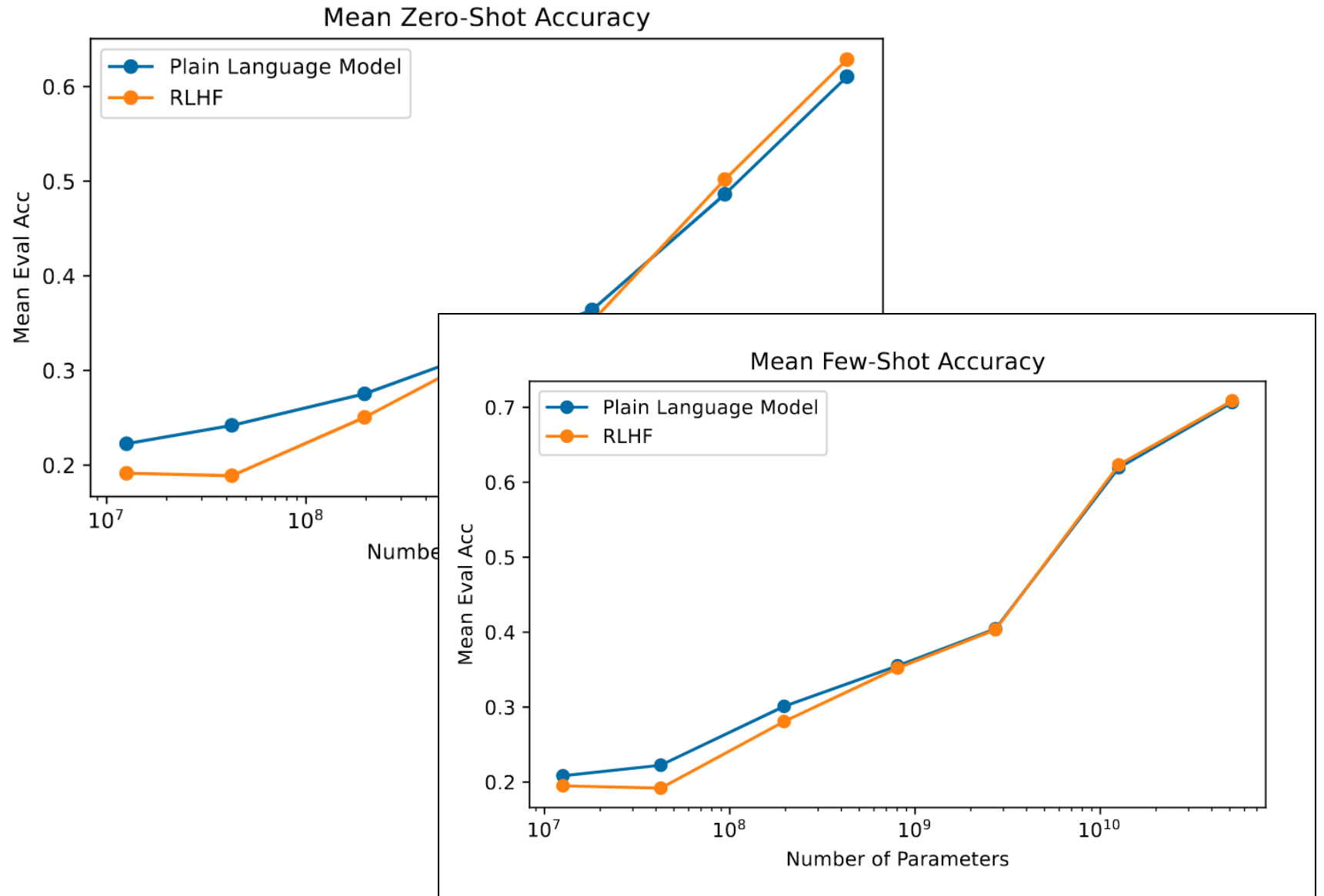
- Does RLHF help?
- **Yes, it increases helpfulness and harmless**
- It does not hurt zero-shot or few-shot performance on most tasks

RLHF
Base



RLHF Results

- Does RLHF help?
- Yes, it increases helpfulness and harmlessness
- **It does not hurt zero-shot or few-shot performance on most tasks**



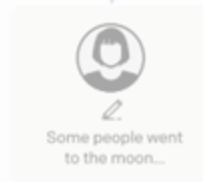
Okay so now
what do we do
with this
thing...?

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.



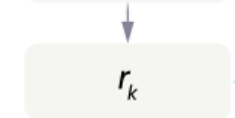
The policy
generates an output.



The reward model
calculates a
reward for the output.



The reward is
used to update
the policy
using PPO.



Reinforcement Learning: Problem Formulation

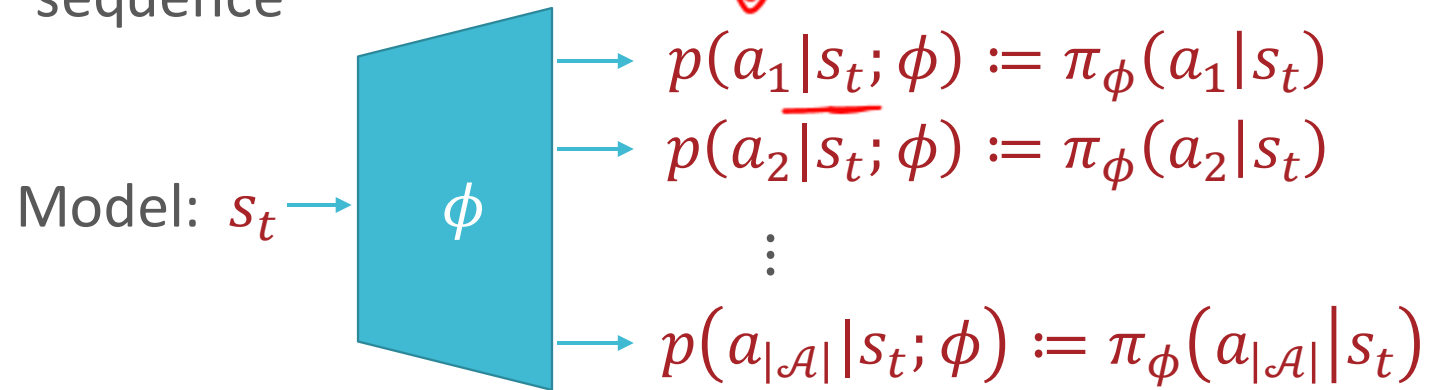
- State space, \mathcal{S} ← prompt x
- Action space, \mathcal{A} ← response y
- Reward function
 - Stochastic, $p(r | s, a)$
 - Deterministic, $R: \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$
- Transition function
 - Stochastic, $p(s' | s, a)$
 - Deterministic, $\delta: \mathcal{S} \times \mathcal{A} \rightarrow \mathcal{S}$

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, $p(r | s, a)$
 - Deterministic reward based on reward model trained on human feedback, $R_\theta(s, a)$
 - 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, $p(s' | s, a)$
 - Deterministic, $\delta(s, a) = [s, a]$

Reinforcement Learning: Object of Interest for Fine-tuning LLMs

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



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

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