Large Language Models (LLMs): From LLMs to Assistants

M. Soleymani
Sharif University of Technology
Spring 2026

Some slides from Yang and Hashimoto lectures, cs224n, Stanford

Outline

- Instruction fine-tuning
- Reinforcement learning from human preferences
- Enhancing LLM applications: Beyond the model

Instruction

- Language models are not aligned with user intent

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.
```

Ouyang et al., Training language models to follow instructions with human feedback, 2022

Language Modeling ≠ Assisting Users

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

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

Explain the theory of relativity to a 6 year old in a few sentences.

Explain the big bang theory to a 6 year old.

Explain evolution to a 6 year old.
```

Language models are not aligned with user intent

Finetuning to the rescue!

Ouyang et al., Training language models to follow instructions with human feedback, 2022

Instruction Tuning

- Finetuning to the rescue!

PROMPT *Explain the moon landing to a 6 year old in a few sentences.*

COMPLETION **Human**
A giant rocket ship blasted off from Earth carrying astronauts to the moon. The astronauts landed their spaceship on the moon and walked around exploring the lunar surface. Then they returned safely back to Earth, bringing home moon rocks to show everyone.

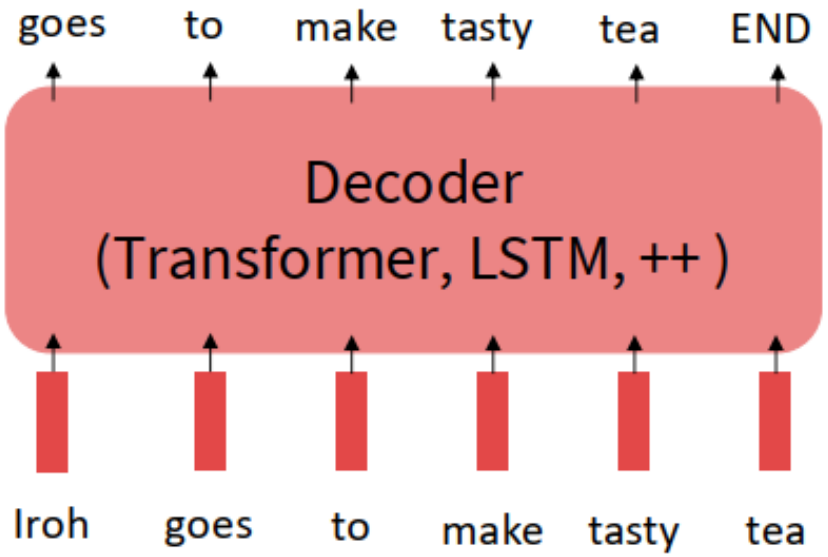
Quyang et al., Training language models to follow instructions with human feedback, 2022

The Pretraining/Finetuning Paradigm

Pretraining can improve NLP applications by serving as parameter initialization.

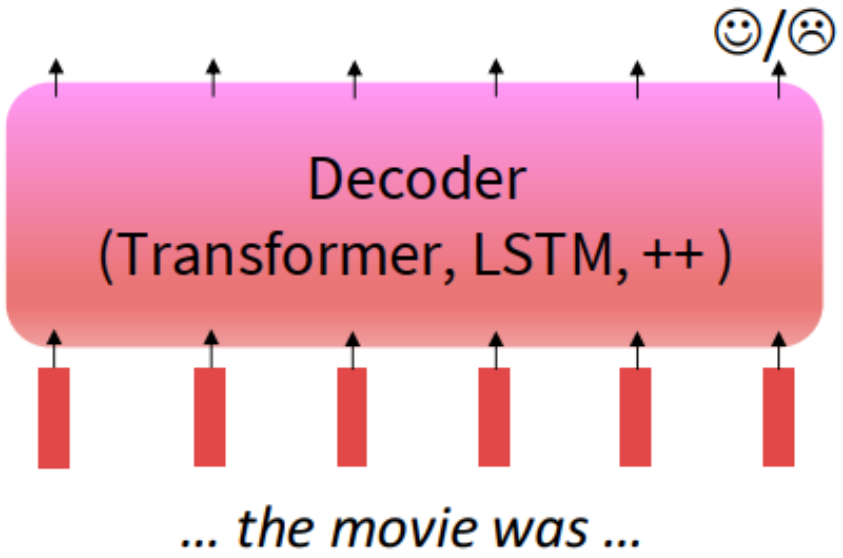
Step 1: Pretrain (on language modeling)

Lots of text; learn general things!



Step 2: Finetune (on your task)

Not many labels; adapt to the task!

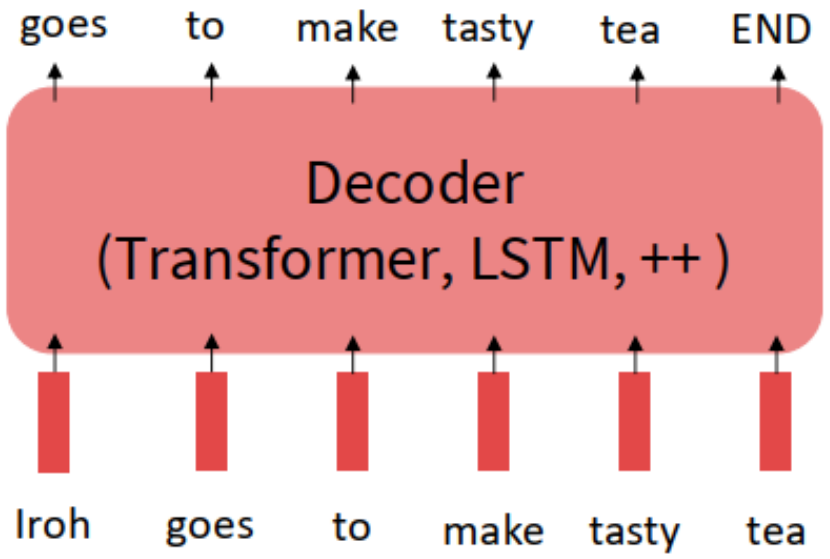


Scaling-up Finetuning

Pretraining can improve NLP applications by serving as parameter initialization.

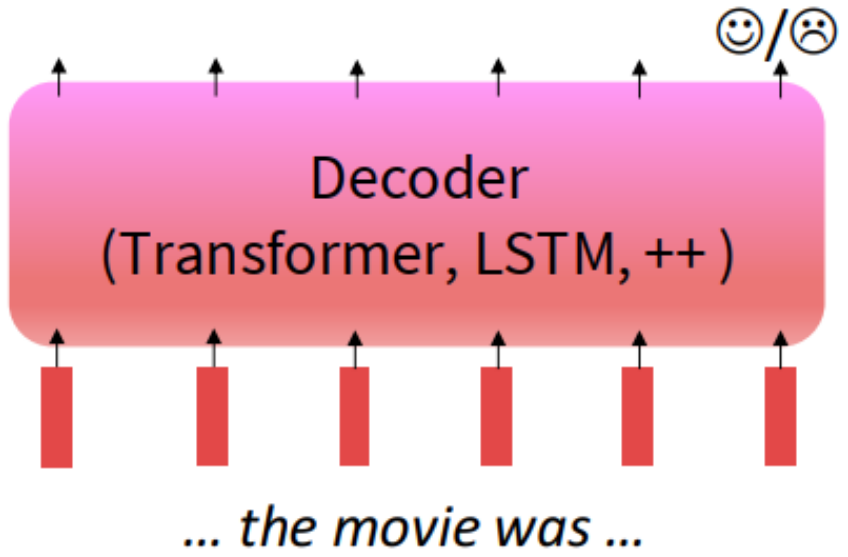
Step 1: Pretrain (on language modeling)

Lots of text; learn general things!

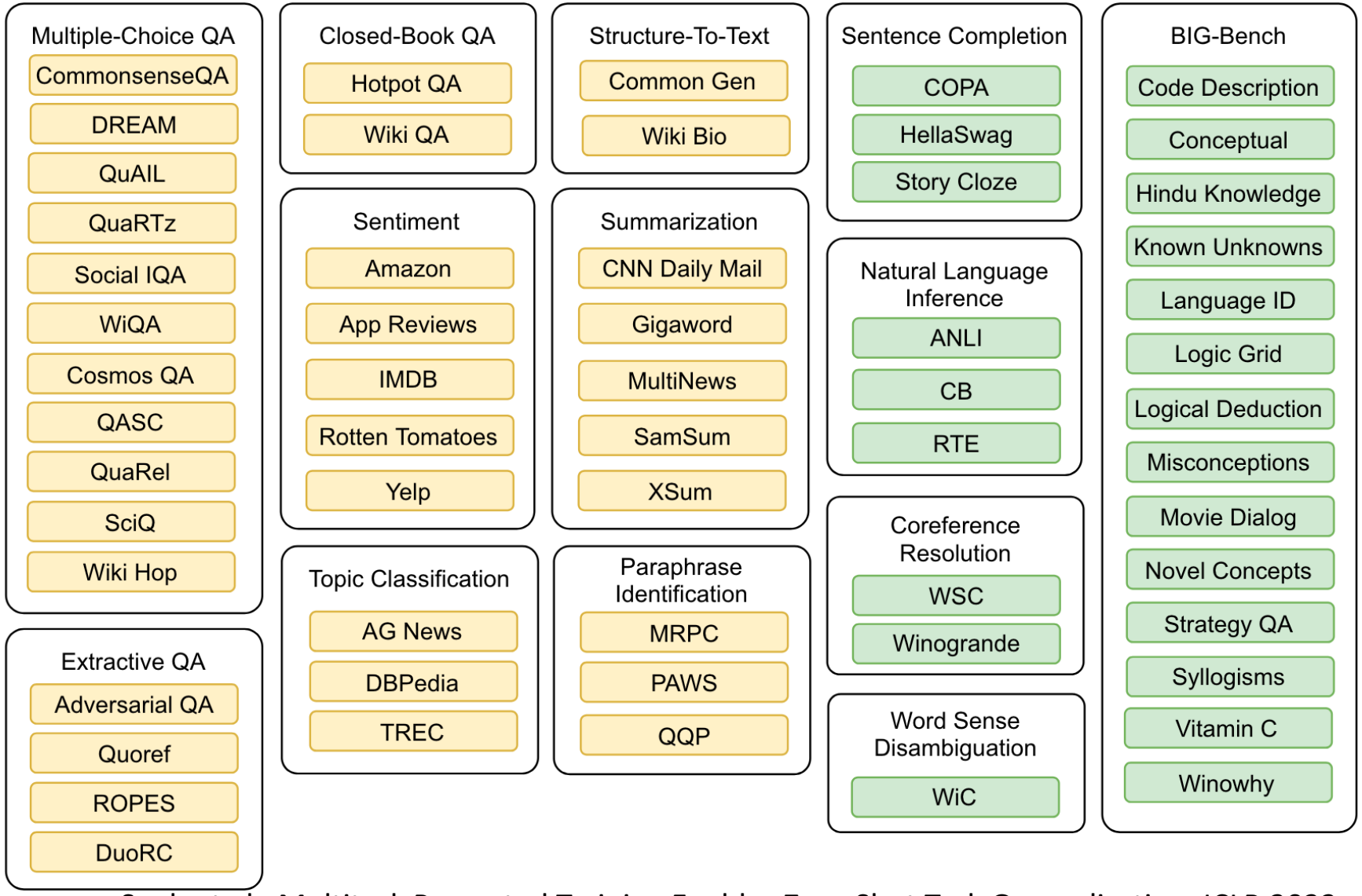


Step 2: Finetune (on many tasks)

~~Not~~ many labels; adapt to the tasks!



T0 Datasets and Task Taxonomy



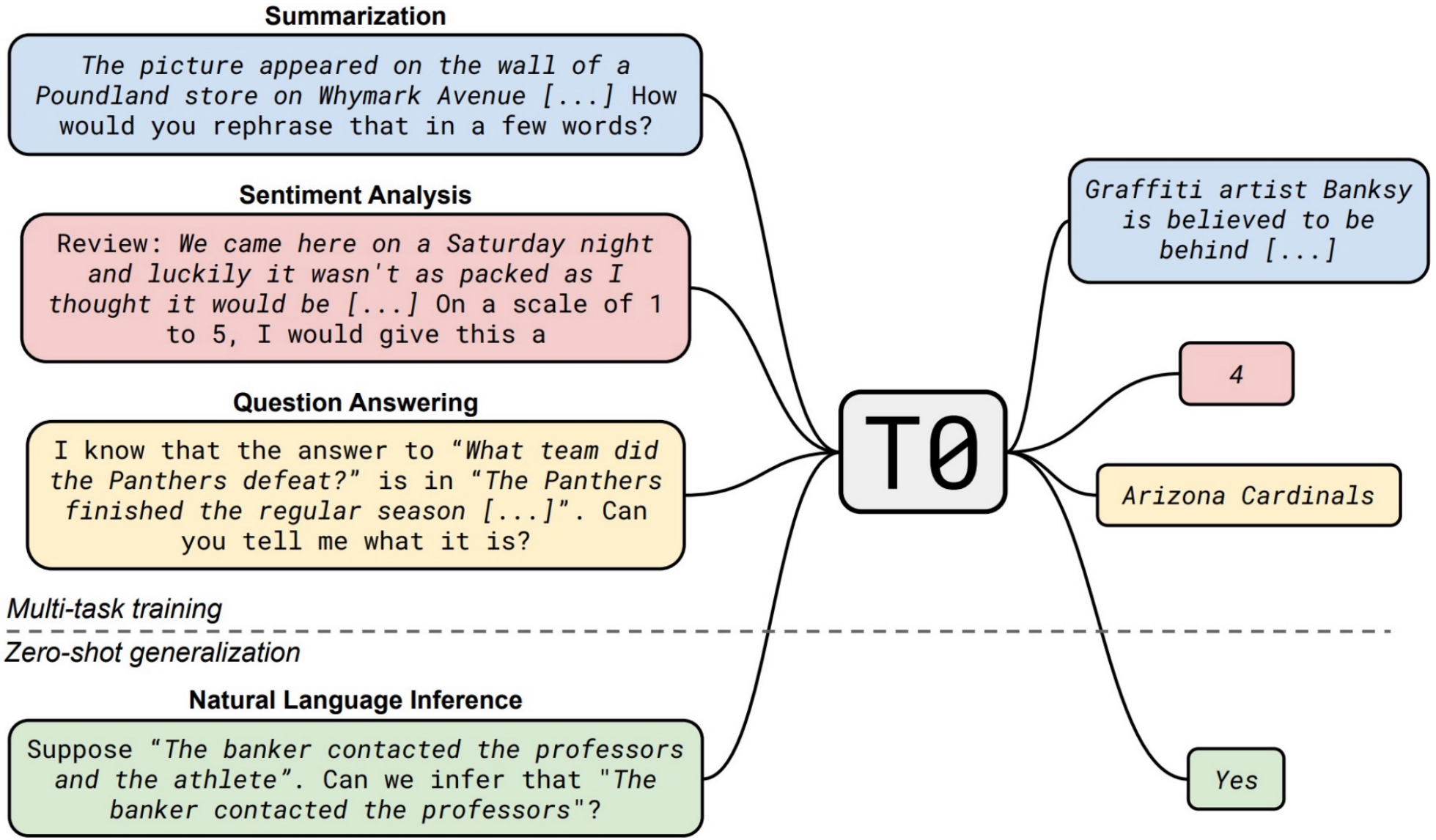
Sanh et al., Multitask Prompted Training Enables Zero-Shot Task Generalization, ICLR 2022

Zero-Shot Learning: T0

- Multitask prompted training improve generalization to held-out tasks
  - Explicit multi-task learning to achieve ZSL.
  - Map any NLP task into a readable prompt
- T0: An encoder-decoder model
- Fine-tuned the T5 model on multi-task training dataset.
- 16x smaller than GPT-3
  - matches or exceeds the performance of GPT-3 on 9 out of 11 held-out datasets

Sanh et al., Multitask Prompted Training Enables Zero-Shot Task Generalization, ICLR 2022

Zero-Shot Learning: T0

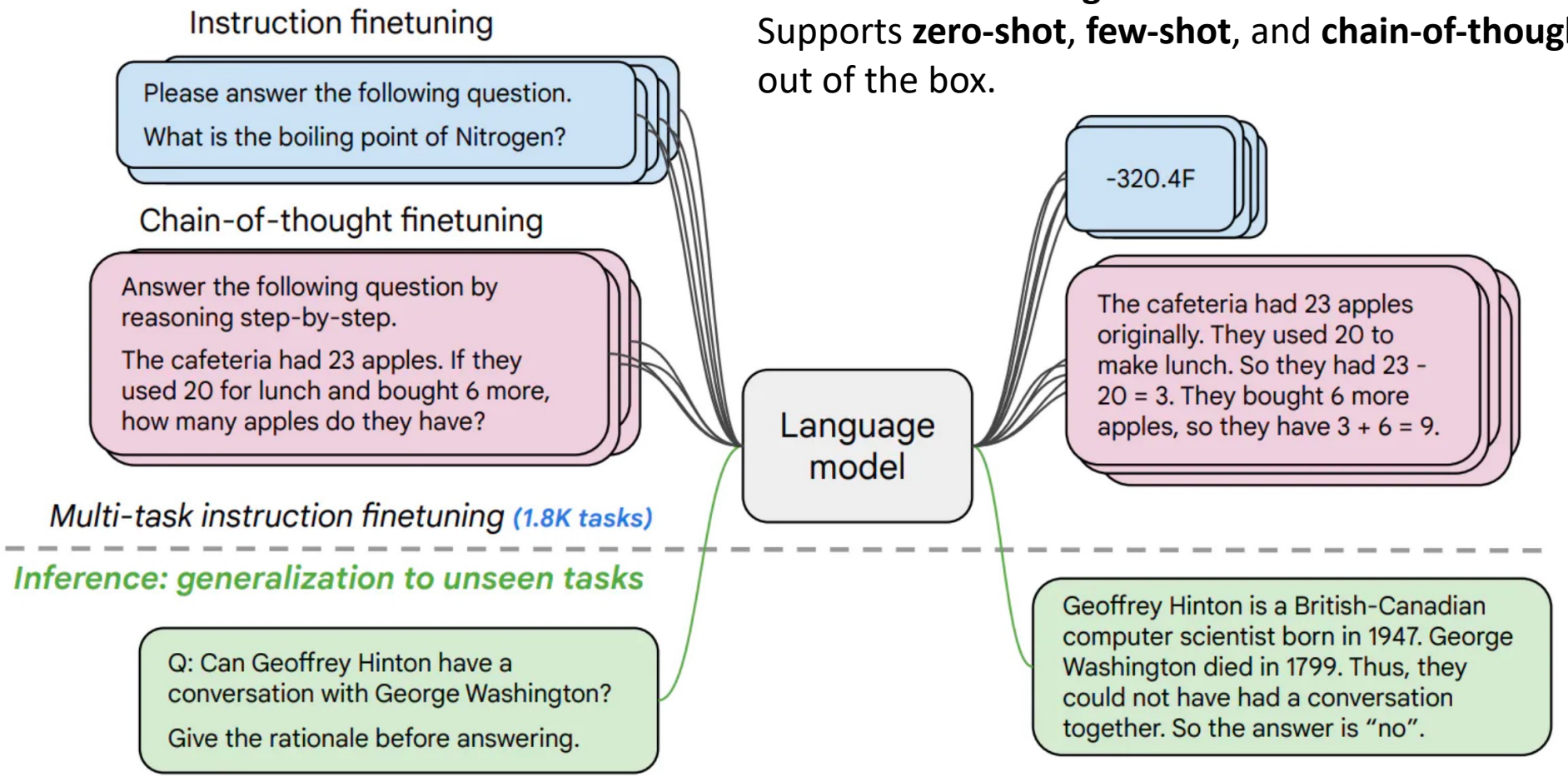


Sanh et al., Multitask Prompted Training Enables Zero-Shot Task Generalization, ICLR 2022

Instruction Finetuning: FLAN

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

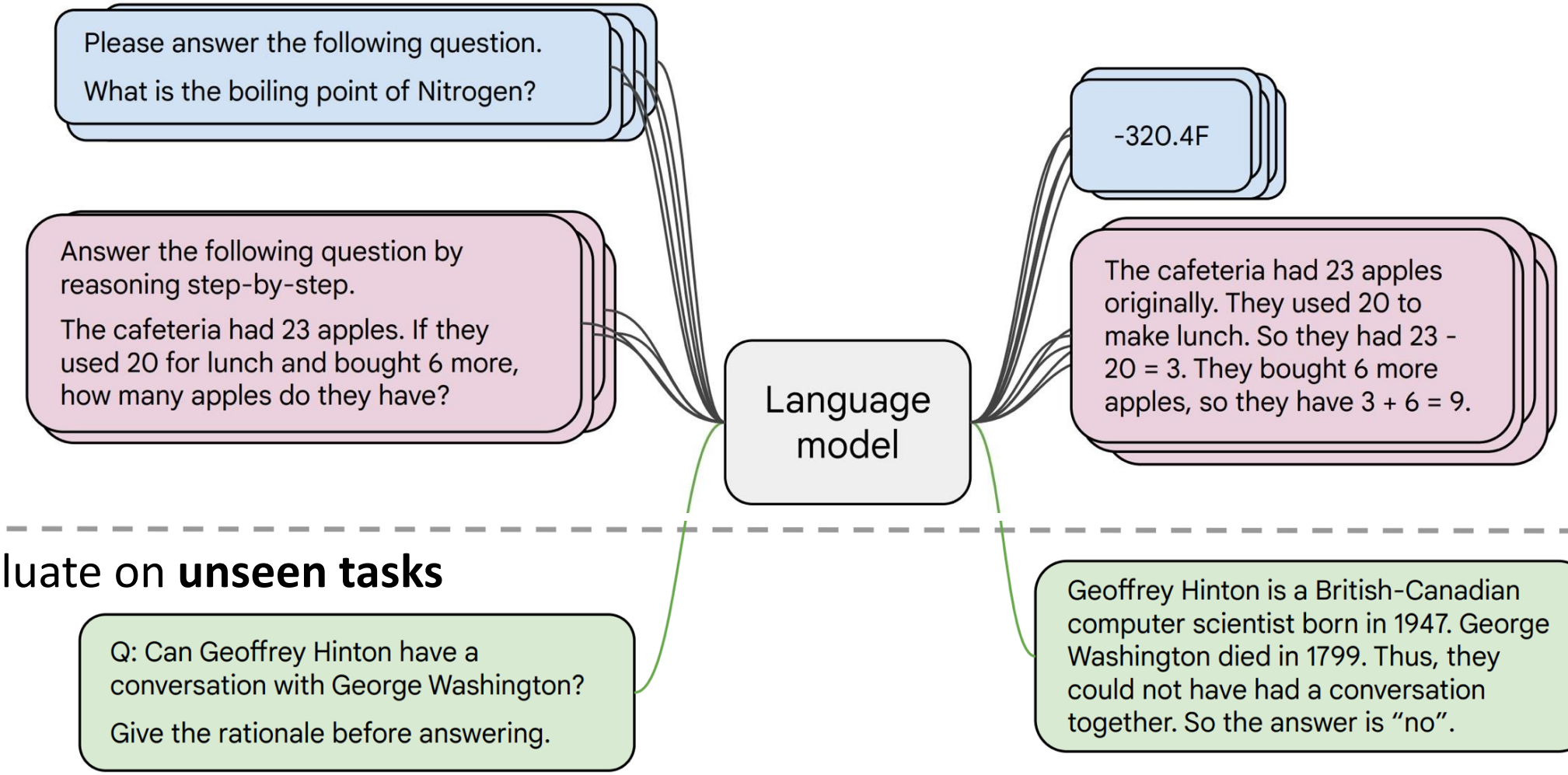
Uses both **machine-generated** and **human-written** instructions.
Supports **zero-shot**, **few-shot**, and **chain-of-thought** prompting out of the box.



Chung et al., Scaling Instruction-Finetuned Language Models, 2022

Instruction Finetuning: FLAN

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



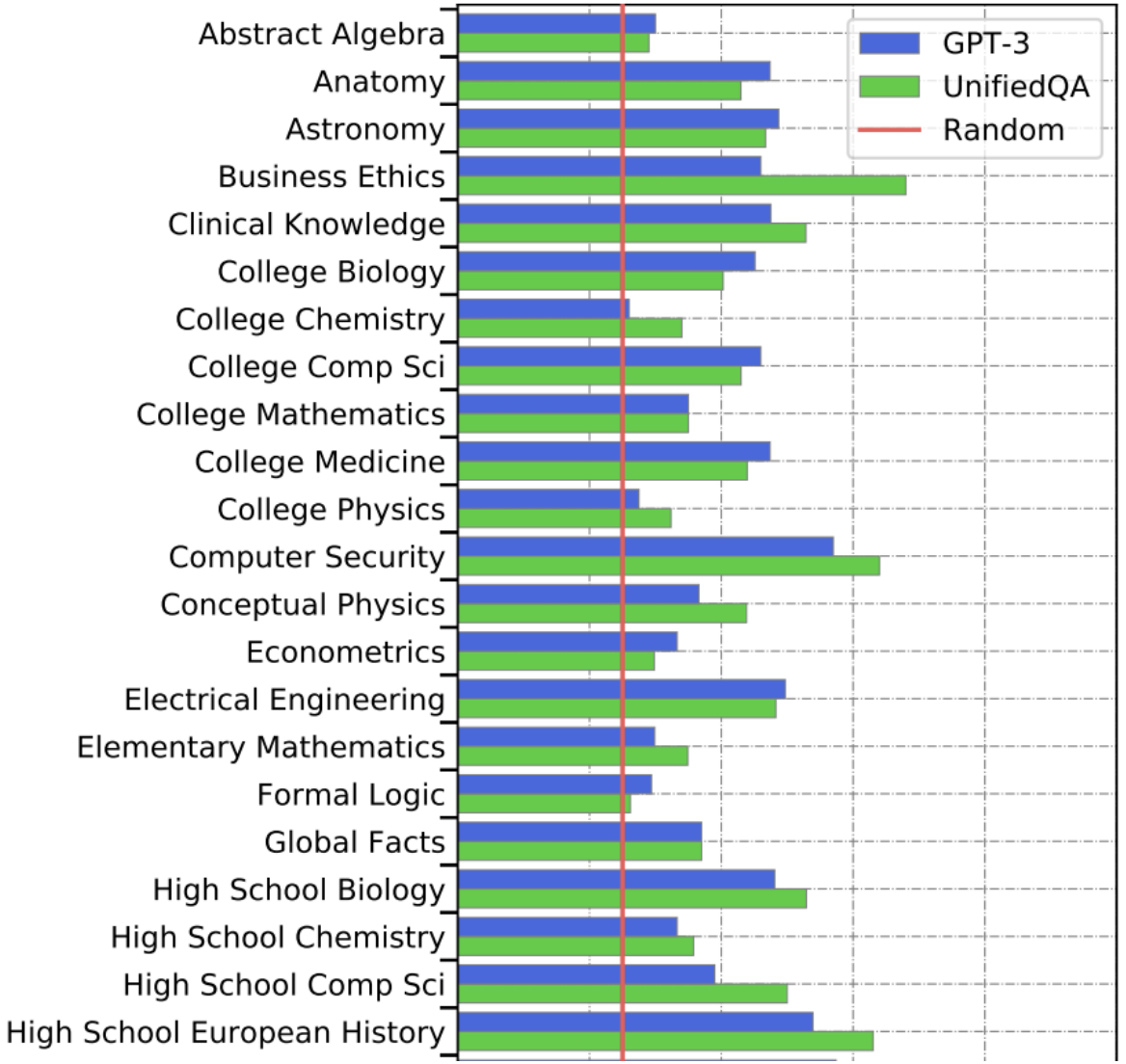
- Evaluate on **unseen tasks**

Chung et al., Scaling Instruction-Finetuned Language Models, 2022

Aside: New Benchmarks for Multitask LMs

Massive Multitask Language Understanding (MMLU)
[Hendrycks et al., 2021]

New benchmarks for measuring LM performance on 57 diverse knowledge intensive tasks



Instruction Finetuning: FLAN (Tasks)

Finetuning tasks

T0-SF

Commonsense reasoning
Question generation
Closed-book QA
Adversarial QA
Extractive QA
Title/context generation
Topic classification
Struct-to-text
...

55 Datasets, 14 Categories, 193 Tasks

Muffin

Natural language inference
Code instruction gen.
Program synthesis
Dialog context generation
Closed-book QA
Conversational QA
Code repair
...

69 Datasets, 27 Categories, 80 Tasks

CoT (Reasoning)

Arithmetic reasoning
Commonsense Reasoning
Implicit reasoning
Explanation generation
Sentence composition
...

9 Datasets, 1 Category, 9 Tasks

Natural Instructions v2

Cause effect classification
Commonsense reasoning
Named entity recognition
Toxic language detection
Question answering
Question generation
Program execution
Text categorization
...

372 Datasets, 108 Categories, 1554 Tasks

- A **Dataset** is an original data source (e.g. SQuAD).
- A **Task Category** is unique task setup (e.g. the SQuAD dataset is configurable for multiple task categories such as extractive question answering, query generation, and context generation).
- A **Task** is a unique <dataset, task category> pair, with any number of templates which preserve the task category (e.g. query generation on the SQuAD dataset.)

Held-out tasks

MMLU

Abstract algebra
College medicine
Professional law
Sociology
Philosophy
...

57 tasks

BBH

Boolean expressions
Tracking shuffled objects
Dyck languages
Navigate
Word sorting
...

27 tasks

TyDiQA

Information seeking QA

8 languages

MGSM

Grade school math problems

10 languages

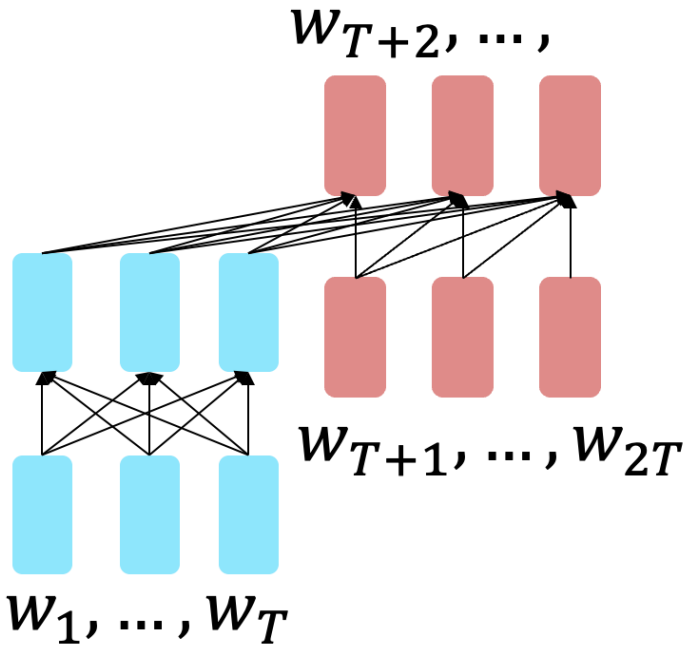
MMLU Results

-	Random	25.0
-	Average human rater	34.5
May 2020	GPT-3 5-shot	43.9
Mar. 2022	Chinchilla 5-shot	67.6
Apr. 2022	PaLM 5-shot	69.3
Oct. 2022	**Flan-PaLM 5-shot**	**72.2**
	Flan-PaLM 5-shot: CoT + SC	**75.2**
-	Average human expert	89.8

Chung et al., Scaling Instruction-Finetuned Language Models, 2022

Instruction Finetuning: FLAN (Performance Gains)

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

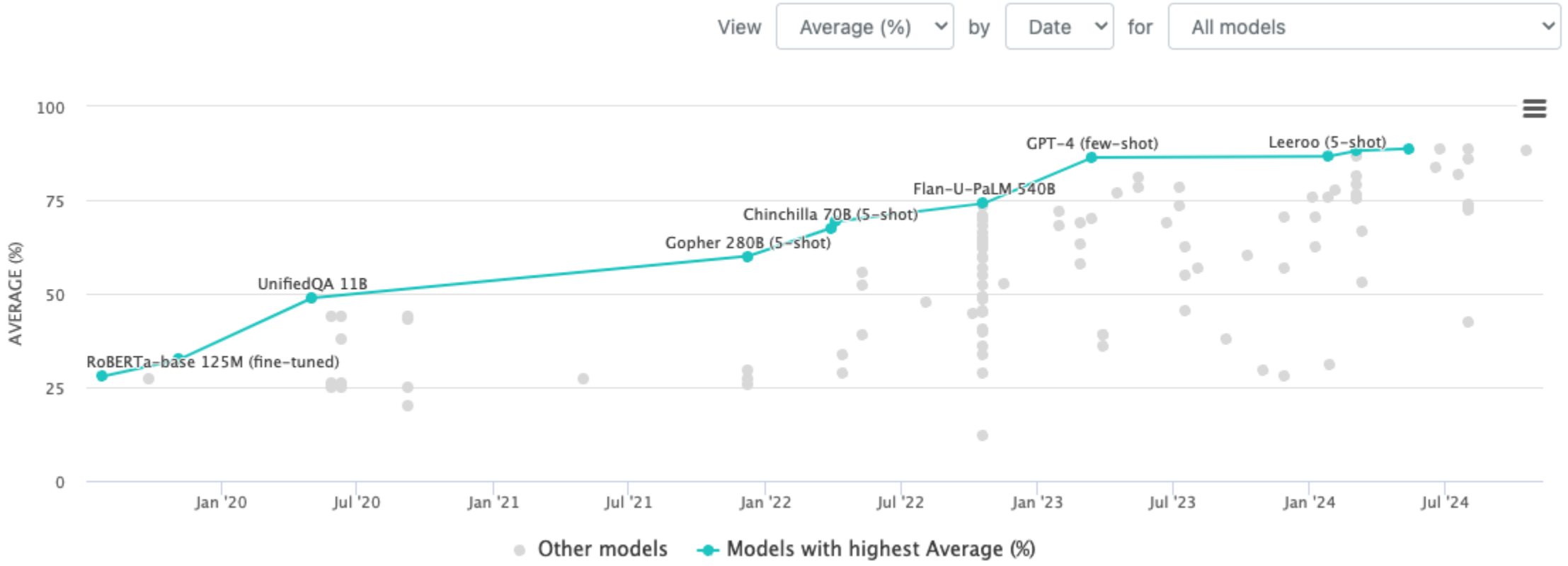


Params	Model	Norm. avg.
80M	T5-Small	-9.2
	Flan-T5-Small	-3.1 (+6.1)
250M	T5-Base	-5.1
	Flan-T5-Base	6.5 (+11.6)
780M	T5-Large	-5.0
	Flan-T5-Large	13.8 (+18.8)
3B	T5-XL	-4.1
	Flan-T5-XL	19.1 (+23.2)
11B	T5-XXL	-2.9
	Flan-T5-XXL	23.7 (+26.6)

Bigger model = bigger Δ

Chung et al., Scaling Instruction-Finetuned Language Models, 2022

Progress on MMLU



- Rapid, impressive progress on challenging knowledge-intensive benchmarks

Example: Before Instruction Tuning

Model input (Disambiguation QA)

Q: In the following sentences, explain the antecedent of the pronoun (which thing the pronoun refers to), or state that it is ambiguous.

Sentence: The reporter and the chef will discuss their favorite dishes.

Options:
(A) They will discuss the reporter's favorite dishes
(B) They will discuss the chef's favorite dishes
(C) Ambiguous

A: Let's think step by step.

Before instruction finetuning

The reporter and the chef will discuss their favorite dishes.
The reporter and the chef will discuss the reporter's favorite dishes.
The reporter and the chef will discuss the chef's favorite dishes.
The reporter and the chef will discuss the reporter's and the chef's favorite dishes.

✖ **(doesn't answer question)**

Example: After Instruction Tuning

Model input (Disambiguation QA)

Q: In the following sentences, explain the antecedent of the pronoun (which thing the pronoun refers to), or state that it is ambiguous.

Sentence: The reporter and the chef will discuss their favorite dishes.

Options:
(A) They will discuss the reporter's favorite dishes
(B) They will discuss the chef's favorite dishes
(C) Ambiguous

A: Let's think step by step.

After instruction finetuning

The reporter and the chef will discuss their favorite dishes does not indicate whose favorite dishes they will discuss. So, the answer is (C). ✅

Try FLAN-T5 out to get a sense of its capabilities: https://huggingface.co/google/flan-t5-xxl [Chung et al., 2022]

InstructGPT: Scaling-up to Tens of Thousands of Tasks

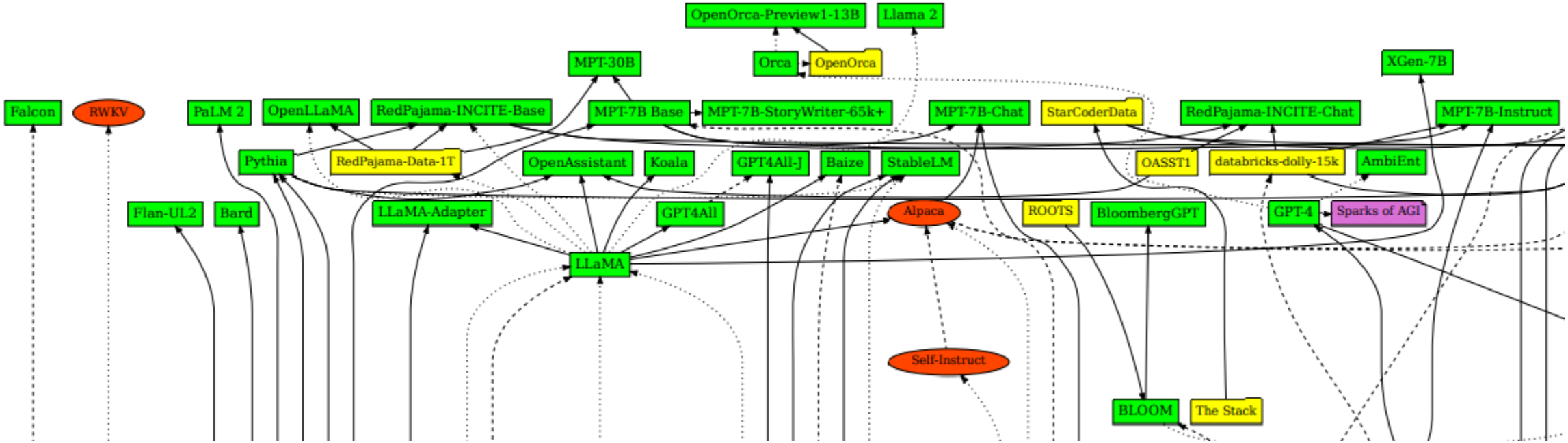
Tasks collected from labelers:

- **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.

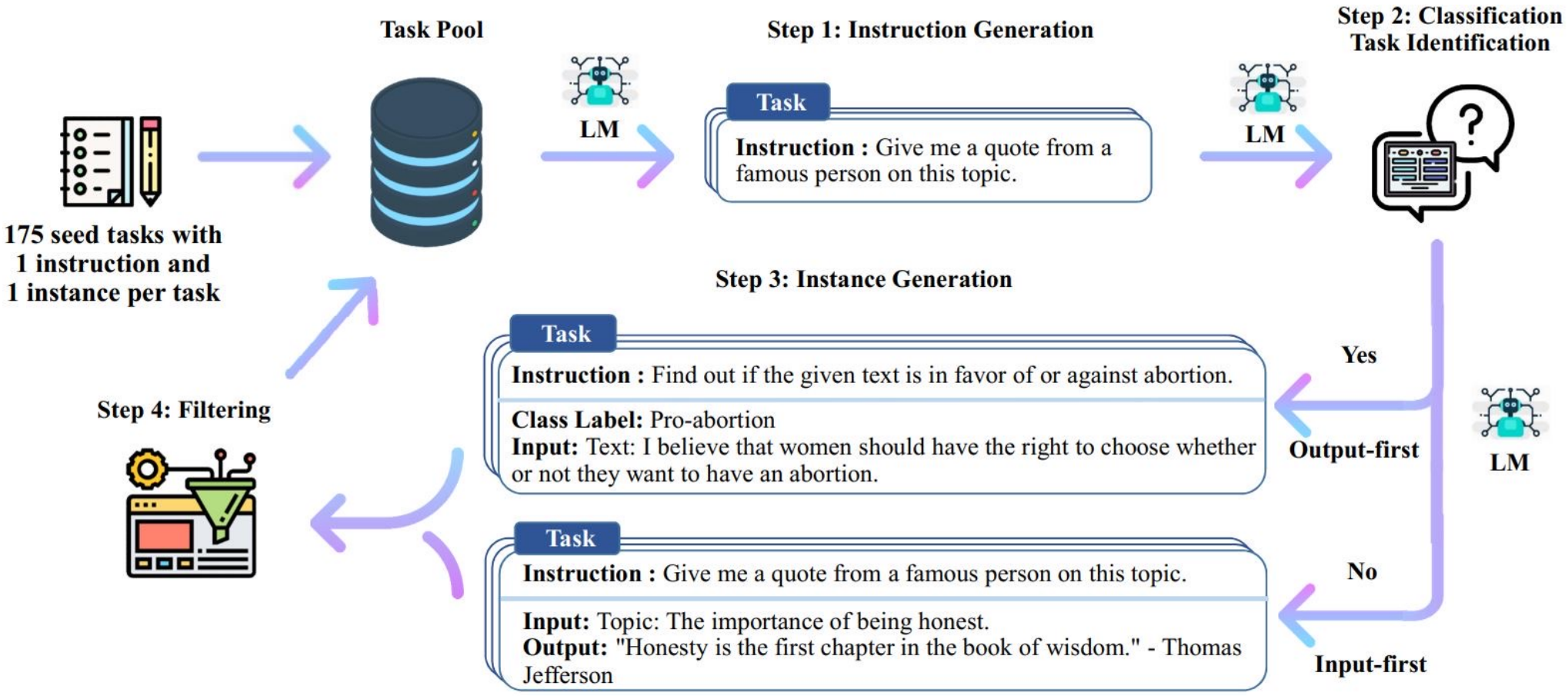
Ouyang et al., Training language models to follow instructions with human feedback, 2022

A Huge Diversity of Instruction-Tuning Datasets



- The release of LLaMA led to open-source attempts to `create' instruction tuning data

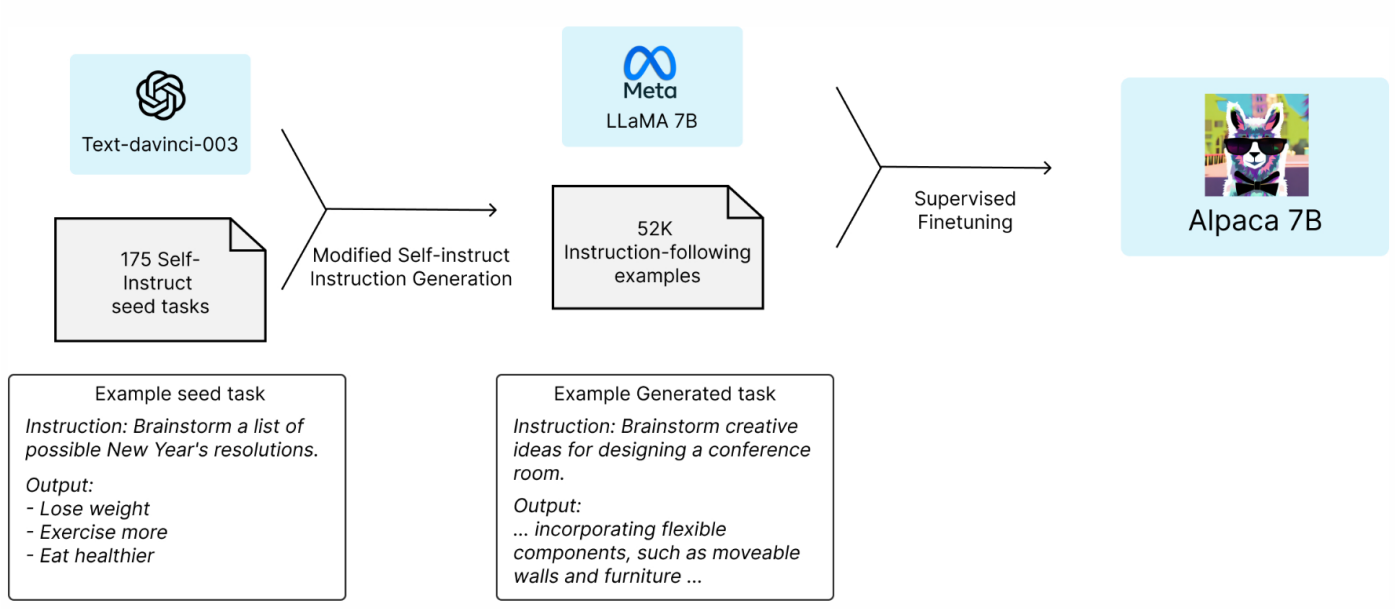
Self-Instruct for Instruction Tuning of LLMs



Wang et al., SELF-INSTRUCT: Aligning Language Models with Self-Generated Instructions, ACL 2023

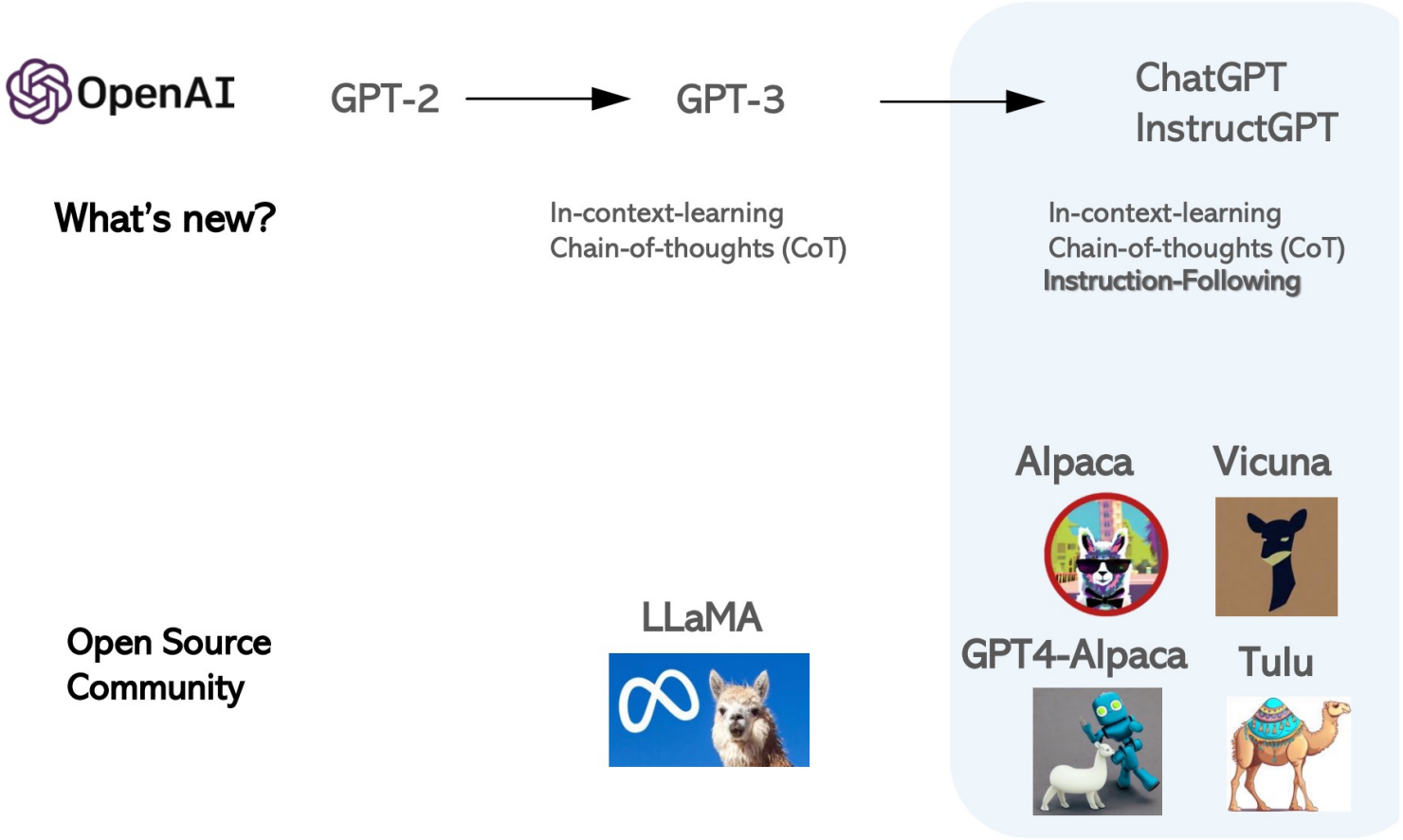
Self-Instruct for Instruction Tuning of LLMs

- Facilitates the preparation of instruction-following examples by employing a semi-automated generation of them
- Example: Alpaca produces 52K high-quality instruction-following demonstrations (using text-davinci-003) to fine-tune LLaMA



Taori et al., Alpaca: A Strong, Replicable Instruction-Following Model, 2023

Self-Instruct in Open-source LLMs



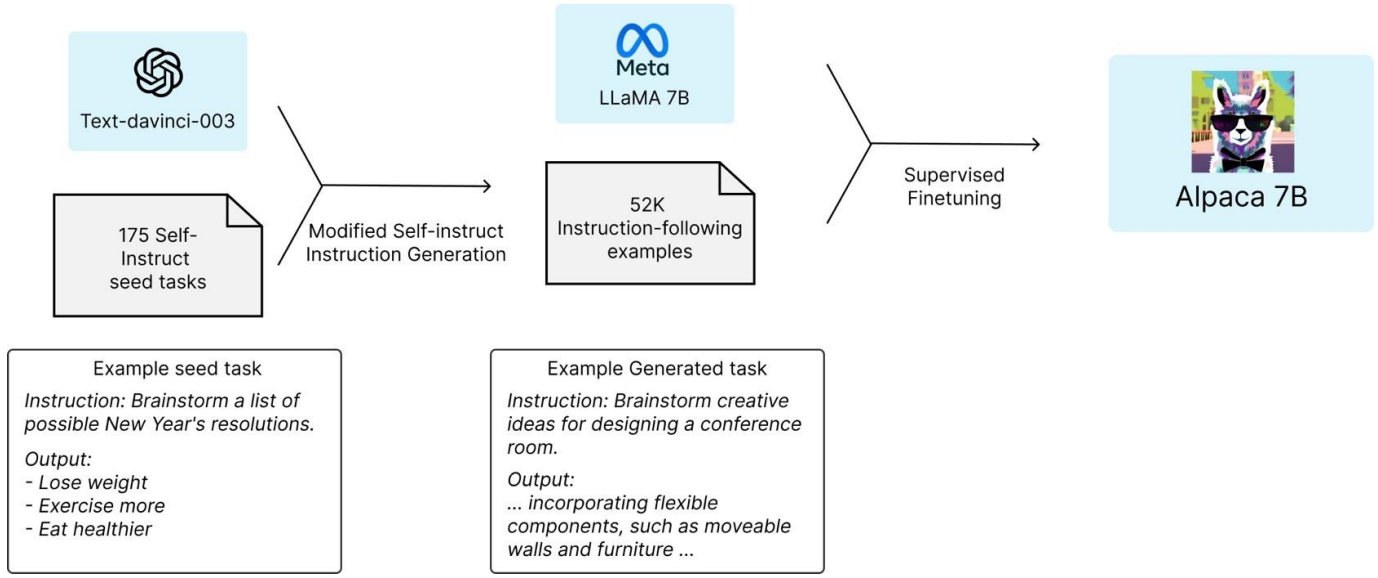
Source: Chunyuan Li's Slides. CVPR 2023 Tutorial

What have we learned from this?

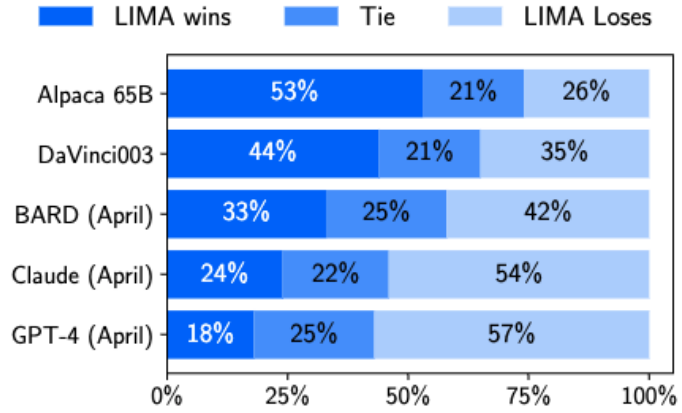
- Generate instructions, input, and output from a LM [Wang et al., 2022]

- **Alpaca**: fine-tuned from the LLaMA 7B model on 52K instruction-following examples

- You don't need many samples to instruction tune (e.g., "LIMA: Less I More for Alignment" Zhou et al., 2023)



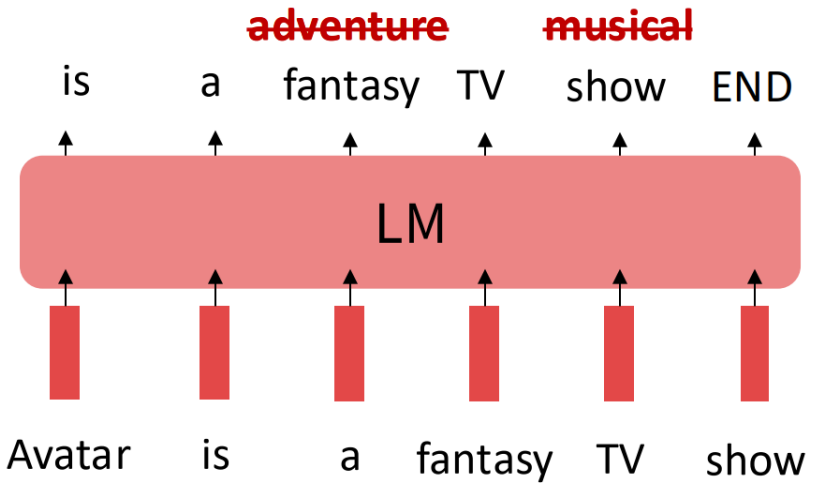
Source	#Examples
Training	
Stack Exchange (STEM)	200
Stack Exchange (Other)	200
wikiHow	200
Pushshift r/WritingPrompts	150
Natural Instructions	50
Paper Authors (Group A)	200



Limitations of Instruction Finetuning?

+ Simple and straightforward, generalize to unseen tasks
- **It's expensive to collect ground-truth data for tasks.**
- **Tasks like open-ended creative generation have no right answer.**
  - Write me a story about a dog and her pet grasshopper.
- **LM penalizes all token-level mistakes equally, but some errors are worse than others.**

- Even with instruction finetuning, there a mismatch between the LM objective and the objective of "satisfy human preferences"!
- Can we **explicitly attempt to satisfy human preferences?**



Outline

- Instruction fine-tuning
- **Reinforcement learning from human preferences**
- Enhancing LLM applications: Beyond the model

Instrucion Tuning and RLHF

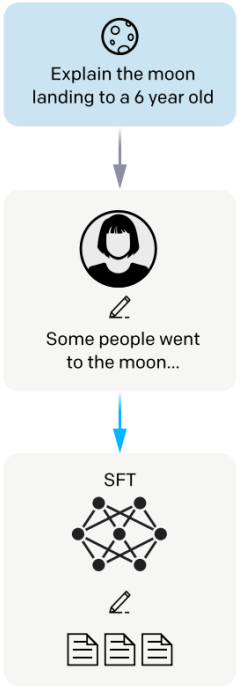
Step 1

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.



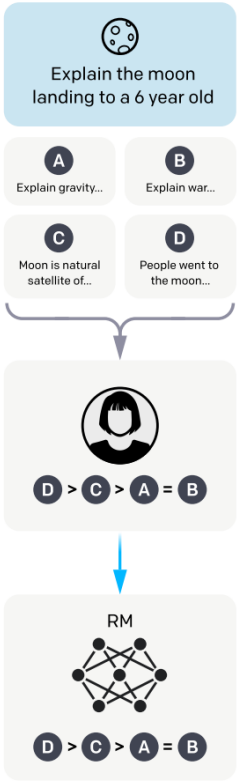
Step 2

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.



Step 3

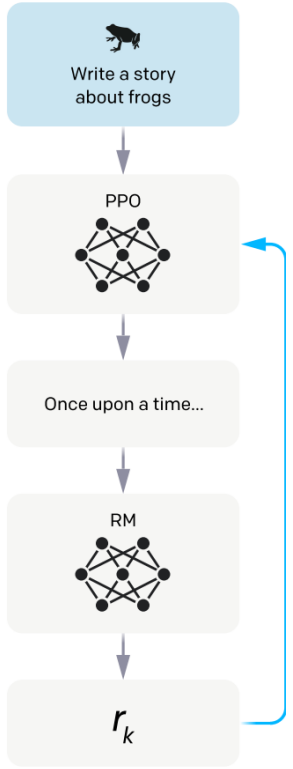
Optimize a policy against the reward model using reinforcement learning.

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.



SFT

Reward model training

RL

Quyang et al., Training language models to follow instructions with human feedback, 2022

Optimizing for Human Preferences

- Let's say we were training a language model on some task (e.g. summarization).
- For each LM 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.
```

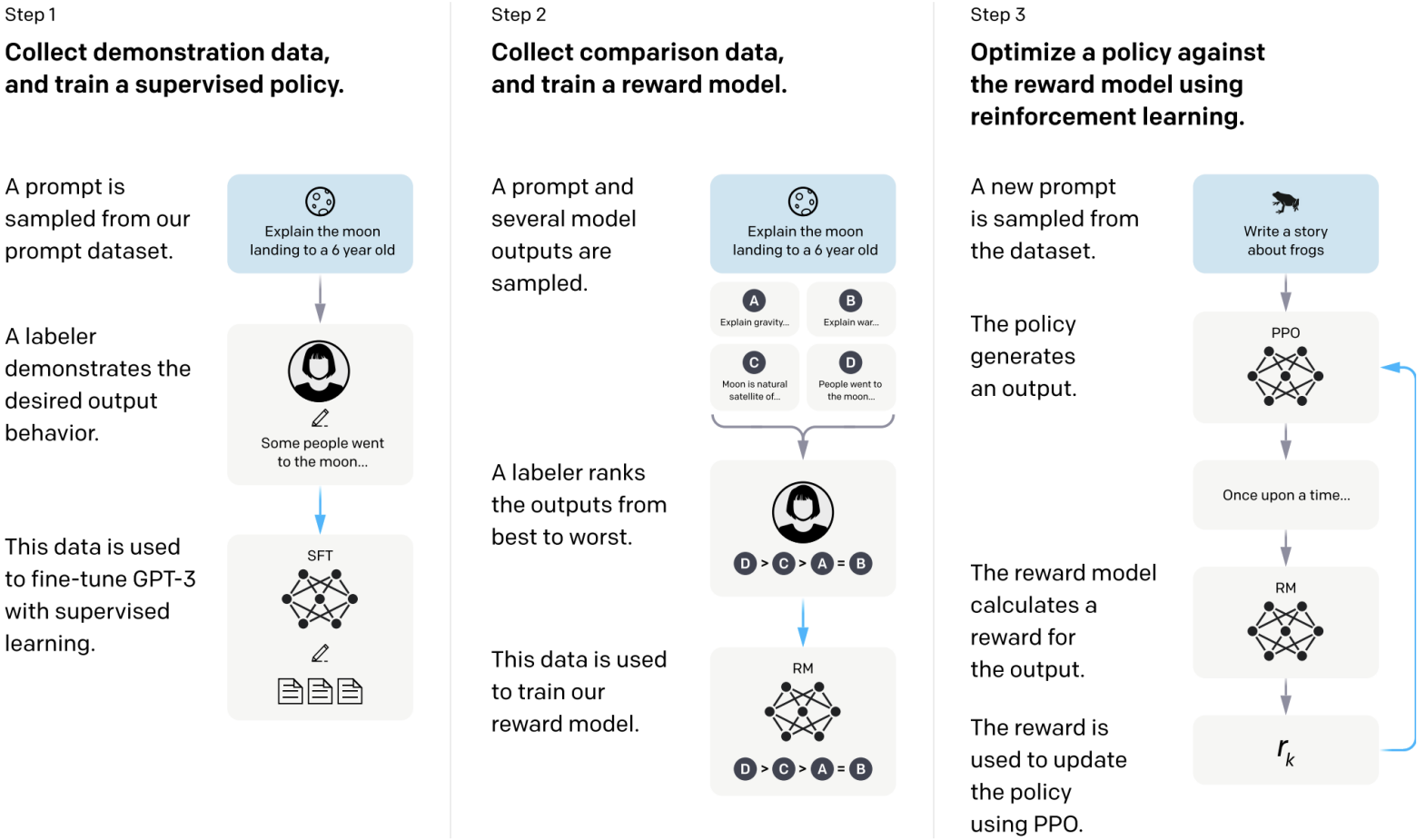
$$s_2$$
$$R(s_2) = 1.2$$

- Now we want to maximize the expected reward of samples from our LM:

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

Note: for mathematical simplicity we're assuming only one "prompt"

High-Level Instantiation: “RLHF” Pipeline



- First step: instruction tuning!
- Second + third steps: maximize reward (but how??)

Optimizing for Human Preferences

- How do we actually change our LM parameters θ to maximize this?

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

- Let's try doing gradient ascent!

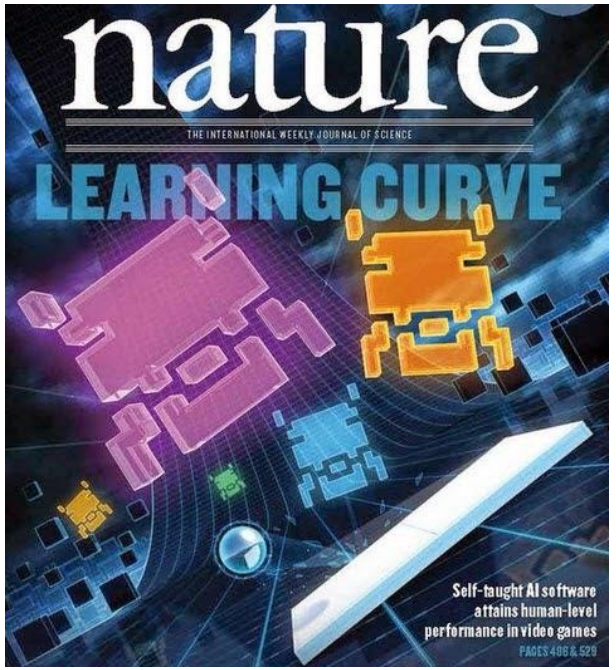
$$\theta_{t+1} \coloneqq \theta_t + \alpha \nabla_{\theta_t} \mathbb{E}_{\hat{s}\sim p_{\theta_t}(s)}[R(\hat{s})]$$

What if our reward function is nondifferentiable??

Reinforcement Learning to the Rescue

- The field of **reinforcement learning (RL)** has studied these (and related) problems for many years now

- But the interest in applying RL to modern LMs is an even newer phenomenon [Ziegler et al., 2019; Stiennon et al., 2020; Ouyang et al., 2022]. **Why?**
  - RL w/ LMs has commonly been viewed as very hard to get right (still is!)
  - Newer advances in RL algorithms that work for large neural models, including language models (e.g. PPO; [Schulman et al., 2017])



AlphaGo

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.
- Not so fast! (Why not?)

- **Problem 1**: human-in-the-loop is expensive!
  - **Solution:** instead of directly asking humans for preferences, **model their preferences** as a separate (NLP) problem! [Knox and Stone, 2009]

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.

$R(s_2) = 1.2$



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 mis-calibrated!

- **Solution:** instead of asking for direct ratings, ask for **pairwise comparisons**, which can be more reliable [Phelps et al., 2015; Clark et al., 2018]

```
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 mis-calibrated!
- **Solution:** instead of asking for direct ratings, ask for **pairwise comparisons**, which can be more reliable [Phelps et al., 2015; Clark et al., 2018]

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.

How Do We Model Human Preferences?

- **Problem 2:** human judgments are noisy and mis-calibrated!
- **Solution:** instead of asking for direct ratings, ask for **pairwise comparisons**, which can be more reliable [Phelps et al., 2015; Clark et al., 2018]

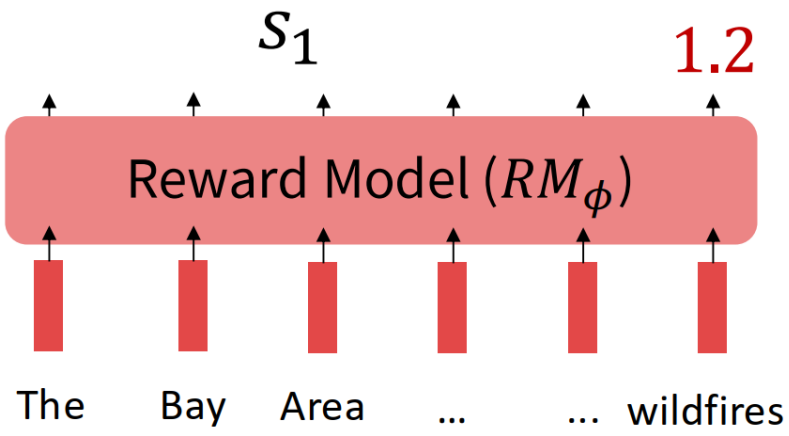
An earthquake hit San Francisco. There was minor property damage, but no injuries.

>

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_3 s_2

Bradley-Terry [1952] paired comparison model

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

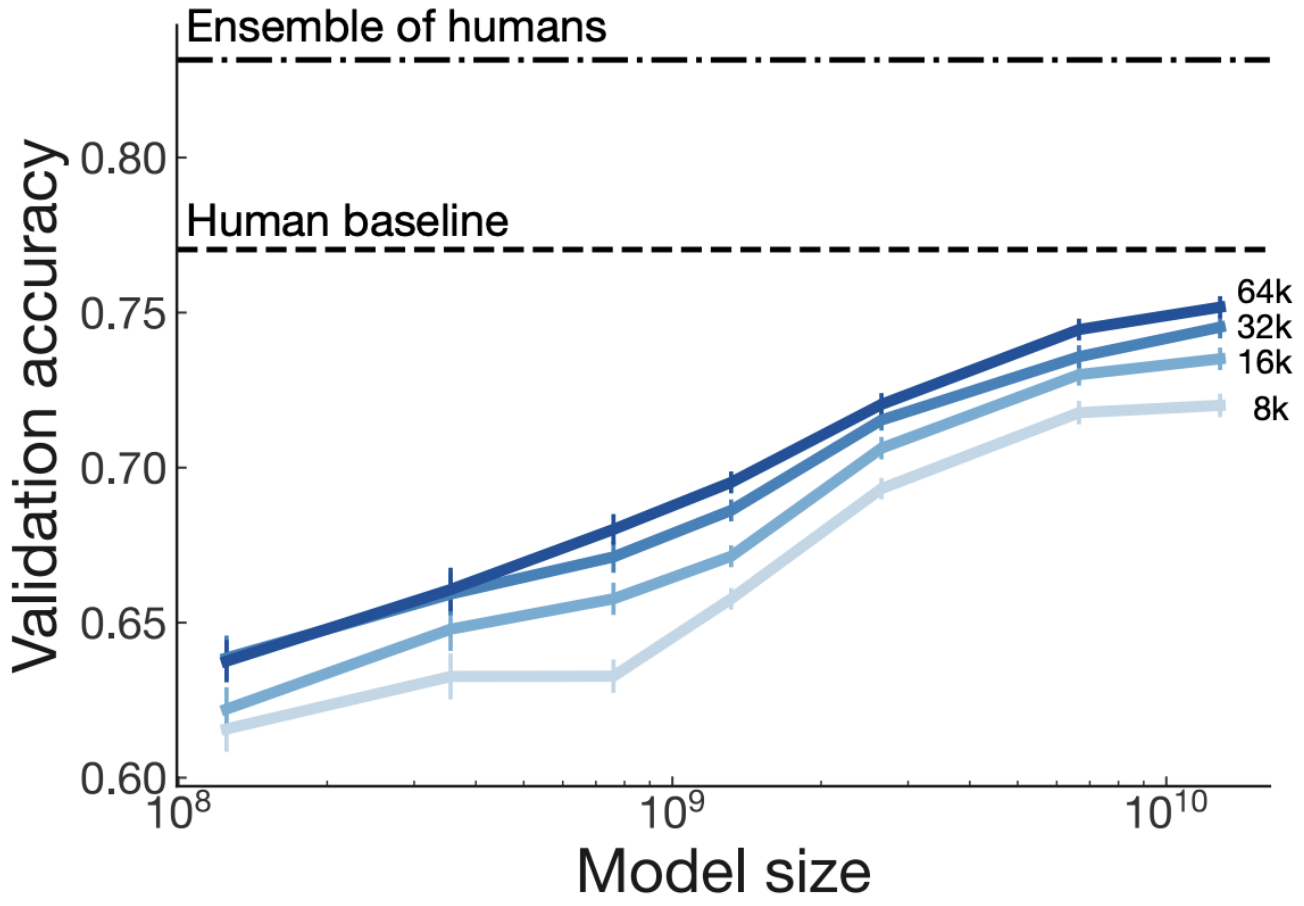
"winning" sample

"losing" sample

s^w should score higher than s^l

Make Sure Your Reward Model Works First!

Evaluate RM on predicting outcome of held-out human judgments

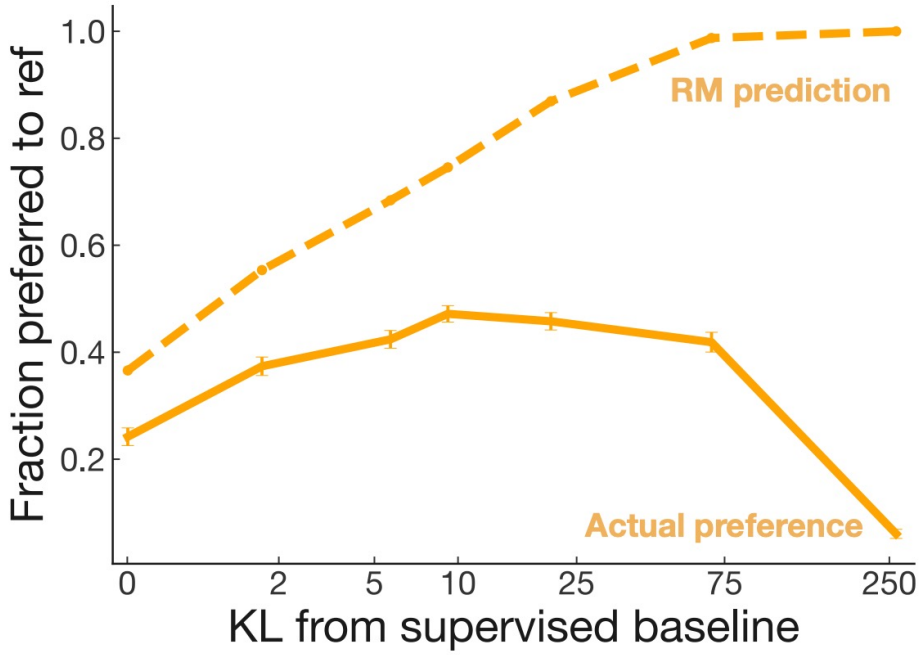


Large enough RM trained on enough data approaching single human perf

Stiennon et al., Learning to summarize from human feedback, NeurIPS 2020

What happens as we optimize the reward model?

- Preference scores versus degree of reward model optimization



Stiennon et al., Learning to summarize from human feedback, NeurIPS 2020

RLHF: Putting It All Together [Christiano et al., 2017; Stiennon et al., 2020]

- 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
  - Optimize the following reward with RL:

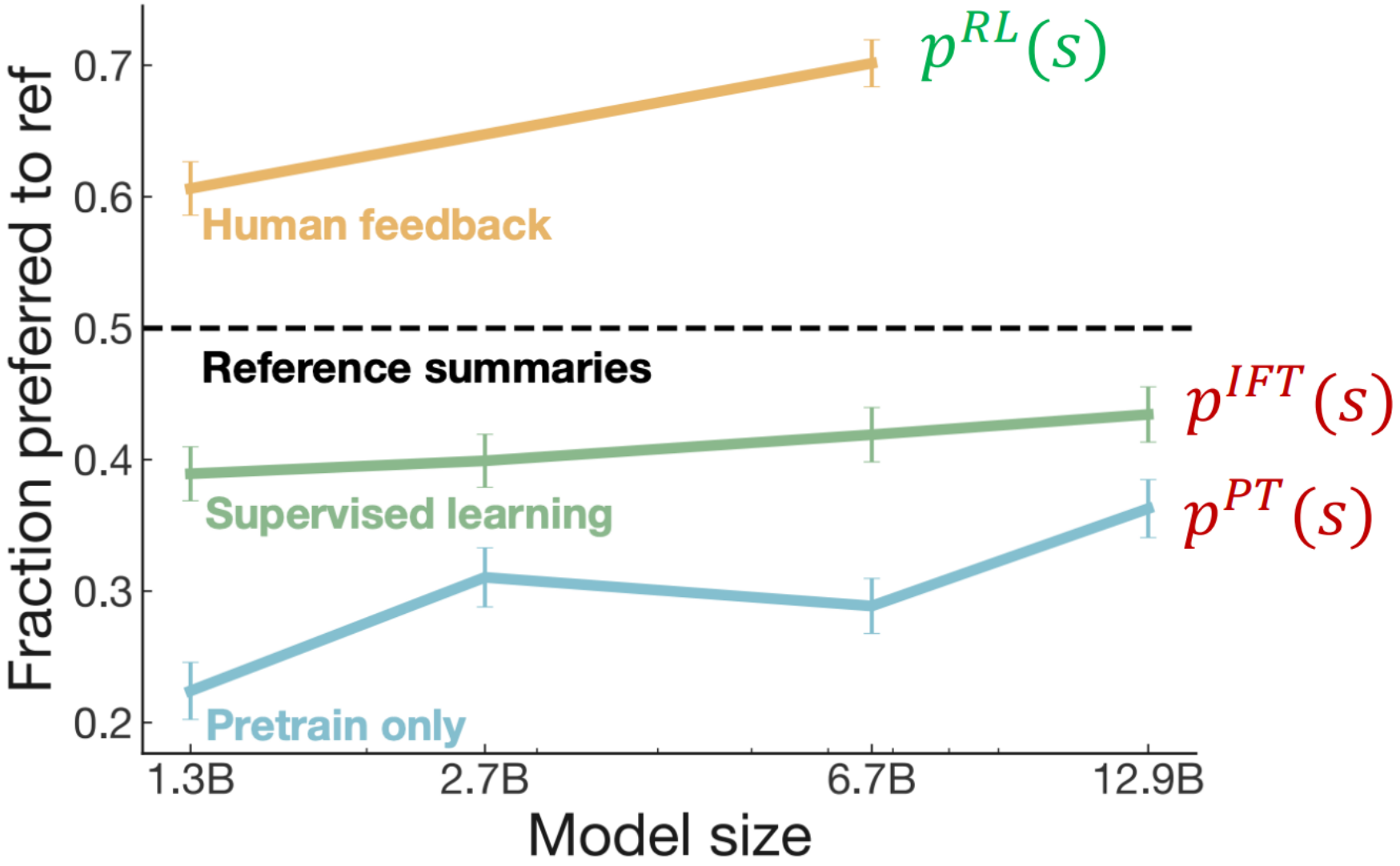
$$R(s) = RM_\phi(s) - \underbrace{\beta \log \left(\frac{p_\theta^{RL}(s)}{p^{PT}(s)} \right)}$$

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

prevents us from diverging too far from the pretrained model (known as the **Kullback-Leibler** (**KL**) divergence between $p_\theta^{RL}(s)$ and $p^{PT}(s)$)

Stiennon et al., Learning to summarize from human feedback, NeurIPS 2020

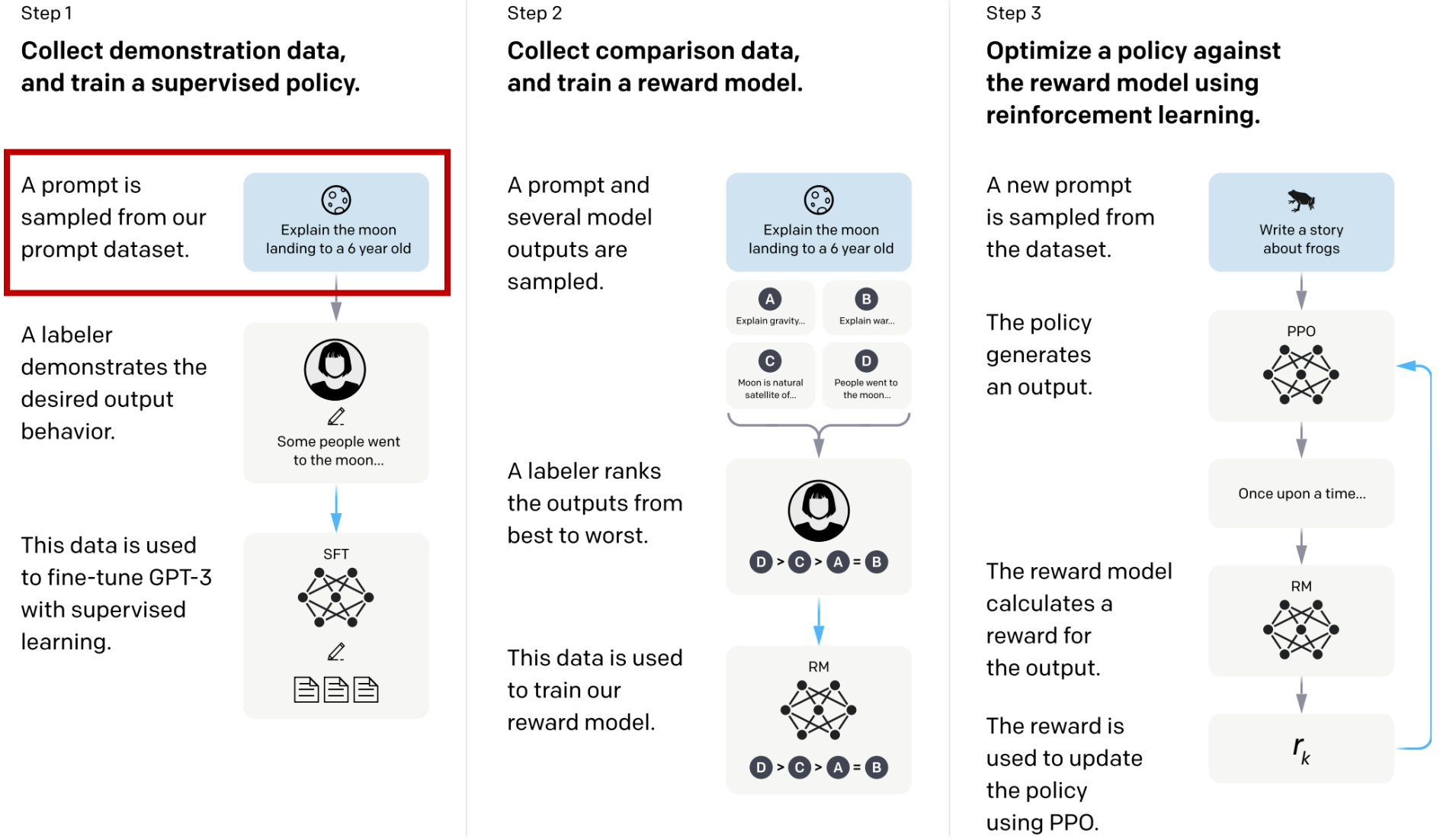
RLHF Provides Gains over Pretraining + Finetuning



Stiennon et al., Learning to summarize from human feedback, NeurIPS 2020

InstructGPT: Scaling-up RLHF to Tens of Thousands of Tasks

30k tasks!



Ouyang et al., Training language models to follow instructions with human feedback, 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.
```

PROMPT *Write a short poem about a wise frog.*

COMPLETION

GPT-3

```
Write a short story in which a character has two different names.

Write a short story in which you try to get something back that you
have lost.

Write a short story in which a character has a bad dream.
```

InstructGPT

```
The frog is a symbol of wisdom
He knows all the secrets of the world
He is a master of disguise
And a great teacher of life
He is a symbol of transformation
And the bringer of change
He is the frog who has seen it all
And knows the meaning of it all
```

ChatGPT: Instruction Finetuning + RLHF for Dialog Agents

ChatGPT: Optimizing Language Models for Dialogue

Note: OpenAI (and similar companies) are keeping **more details secret** about ChatGPT training (including data, training parameters, model size)— perhaps to keep a competitive edge…

Methods

We trained this model using Reinforcement Learning from Human Feedback (RLHF), using the same methods as InstructGPT, but with slight differences in the data collection setup. We trained an initial model using supervised fine-tuning: human AI trainers provided conversations in which they played both sides—the user and an AI assistant. We gave the trainers access to model-written suggestions to help them compose their responses. We mixed this new dialogue dataset with the InstructGPT dataset, which we transformed into a dialogue format.

(Instruction finetuning!)

ChatGPT: Instruction Finetuning + RLHF for Dialog Agents

ChatGPT: Optimizing Language Models for Dialogue

Note: OpenAI (and similar companies) are keeping more details secret about ChatGPT training (including data, training parameters, model size)— perhaps to keep a competitive edge…

Methods

To create a reward model for reinforcement learning, we needed to collect comparison data, which consisted of two or more model responses ranked by quality. To collect this data, we took conversations that AI trainers had with the chatbot. We randomly selected a model-written message, sampled several alternative completions, and had AI trainers rank them. Using these reward models, we can fine-tune the model using Proximal Policy Optimization. We performed several iterations of this process.

(RLHF!)

Limitations of RL + Reward Modeling

- Human preferences are unreliable!
  - Chatbots are rewarded to produce responses that *seem* authoritative and helpful, *regardless of truth*
  - This can result in making up facts + hallucinations

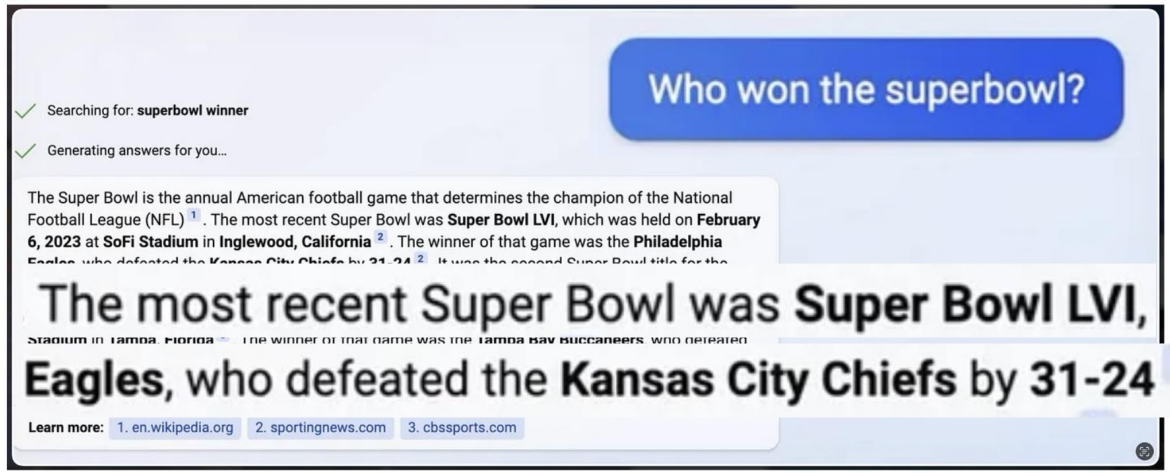
TECHNOLOGY

Google shares drop $100 billion after its new AI chatbot makes a mistake

February 9, 2023 · 10:15 AM ET

https://www.npr.org/2023/02/09/1155650909/google-chatbot-error-bard-shares

Bing AI hallucinates the Super Bowl



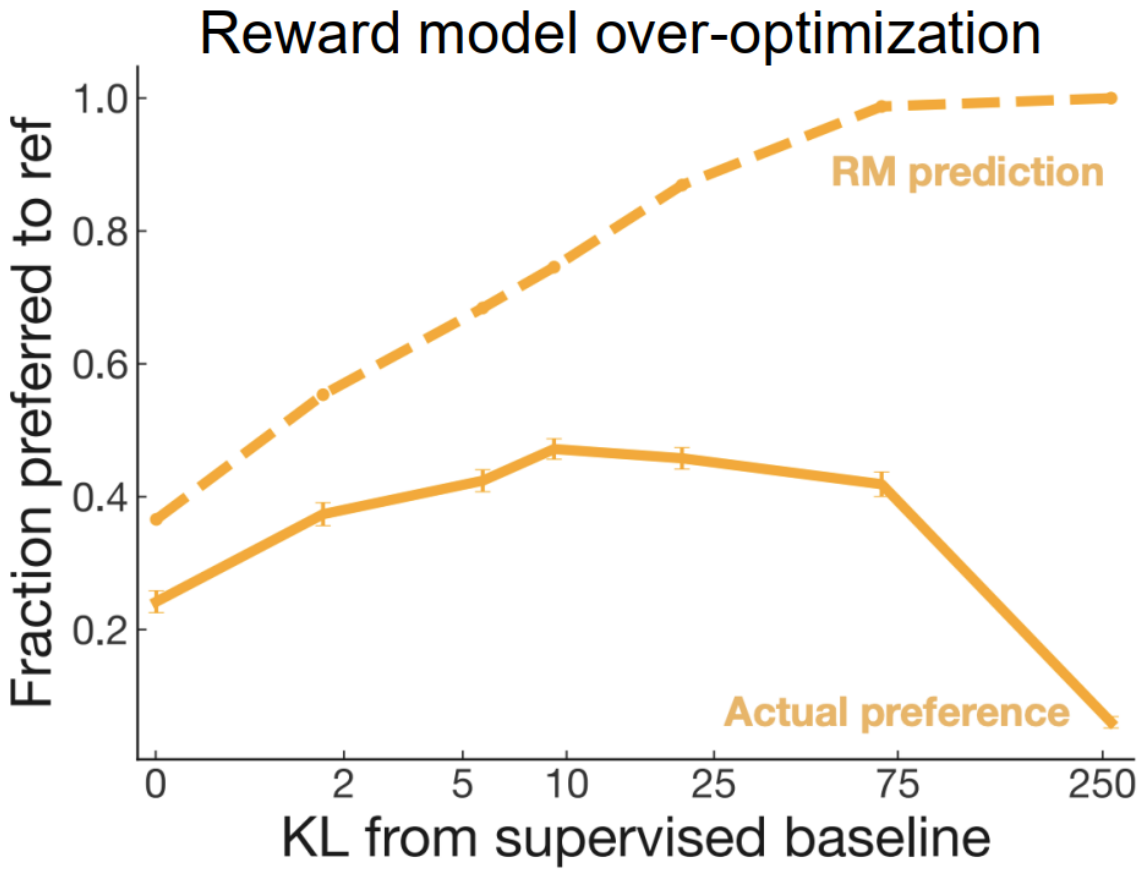
https://news.ycombinator.com/item?id=34776508

https://apnews.com/article/kansas-city-chiefs-philadelphia-eagles-technologyscience-82bc20f207e3e4cf81abc6a5d9e6b23a

Limitations of RL + Reward Modeling

- Human preferences are unreliable!
  - Chatbots are rewarded to produce responses that *seem* authoritative and helpful, *regardless of truth*
  - This can result in making up facts + hallucinations

- **Models** of human preferences are even more unreliable!
  - ”Reward hacking” is a common problem in RL



$$R(s) = RM_\phi(s) - \beta \log \left(\frac{p_\theta^{RL}(s)}{p^{PT}(s)} \right)$$

[Stiennon et al., 2020]

Removing Reward Modeling from RLHF

Recall we want to maximize the following objective in RLHF

$$\mathbb{E}_{\hat{y}\sim p_\theta^{RL}(\hat{y}|x)}[RM_\phi(x,\hat{y}) - \beta \log\left(\frac{p_\theta^{RL}(\hat{y}|x)}{p^{PT}(\hat{y}|x)}\right)]$$

There is a closed form solution to this:

$$p^*(\hat{y}|x) = \frac{1}{Z(x)} p^{PT}(\hat{y}|x) \exp(\frac{1}{\beta} RM(x,\hat{y}))$$

- Rearrange this via a log transformation

$$RM(x,\hat{y}) = \beta\,(\log p^*(\hat{y}|x) - \log p^{PT}(\hat{y}|x)) + \beta \log Z(x) = \beta \log \frac{p^*(\hat{y}|x)}{p^{PT}(\hat{y}|x)} + \beta \log Z(x)$$

- This holds true for anv arbitrarv LMs. Thus

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

Putting it together: Direct Preference Optimization (DPO)

- Derived reward mo $RM_\theta(x, \hat{y}) = \beta \log \frac{p_\theta^{RL}(\hat{y}|x)}{p^{PT}(\hat{y}|x)} + \beta \log Z(x)$

- Final DPO loss via the Bradley-Terry model of human preference

$$J_{DPO}(\theta) = -\mathbb{E}_{(x,y_w,y_l)\sim D}[\log \sigma(RM_\theta(x, y_w) - RM_\theta(x, y_l))]$$

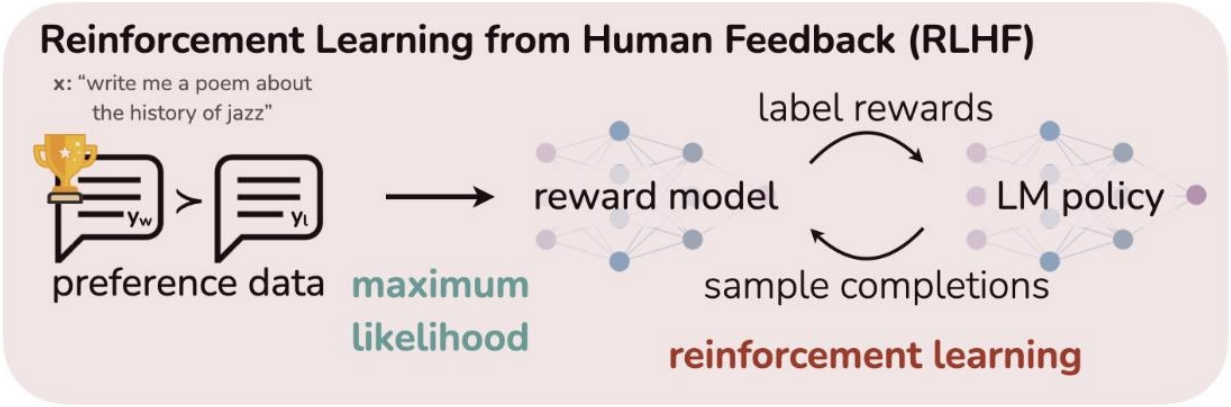
$$= -\mathbb{E}_{(x,y_w,y_l)\sim D}\left[\log \sigma(\beta \log \frac{p_\theta^{RL}(y_w|x)}{p^{PT}(y_w|x)} - \beta \log \frac{p_\theta^{RL}(y_l|x)}{p^{PT}(y_l|x)})\right]$$

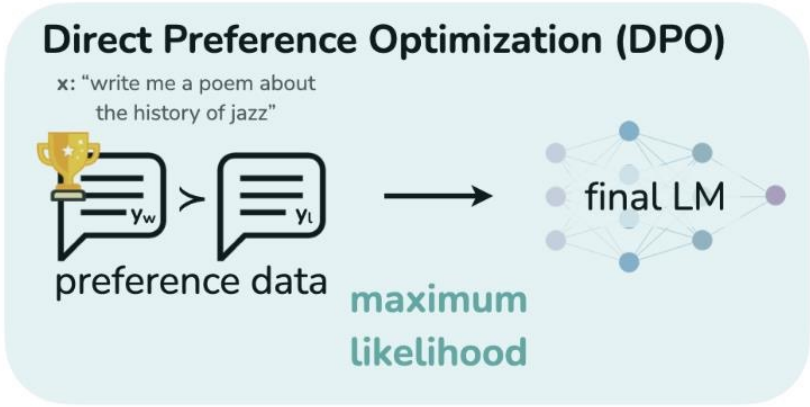
Log Z term cancels as the loss only measures differences in rewards

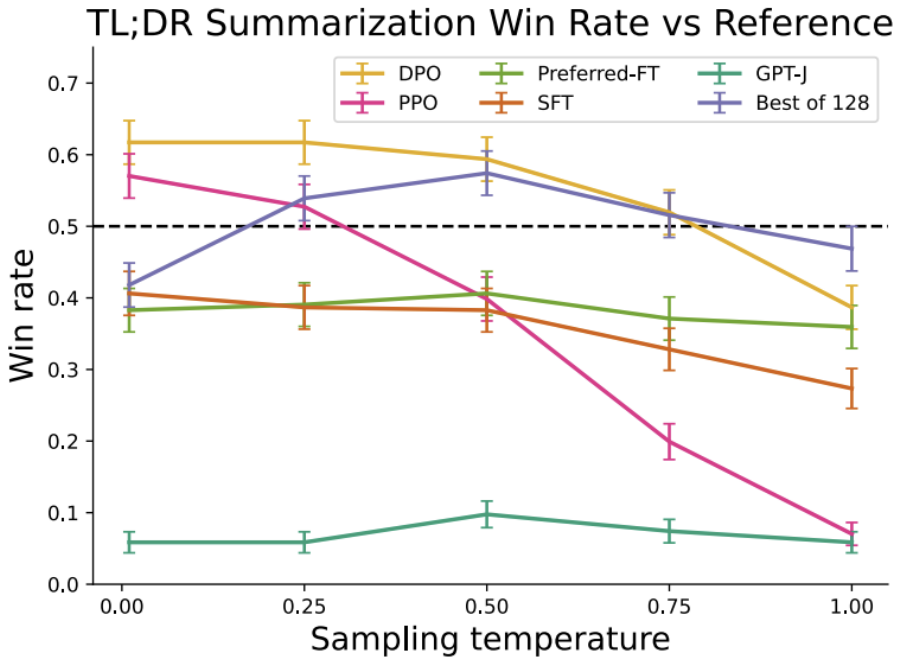
Reward for winning sample

Reward for losing sample

DPO outperforms prior methods

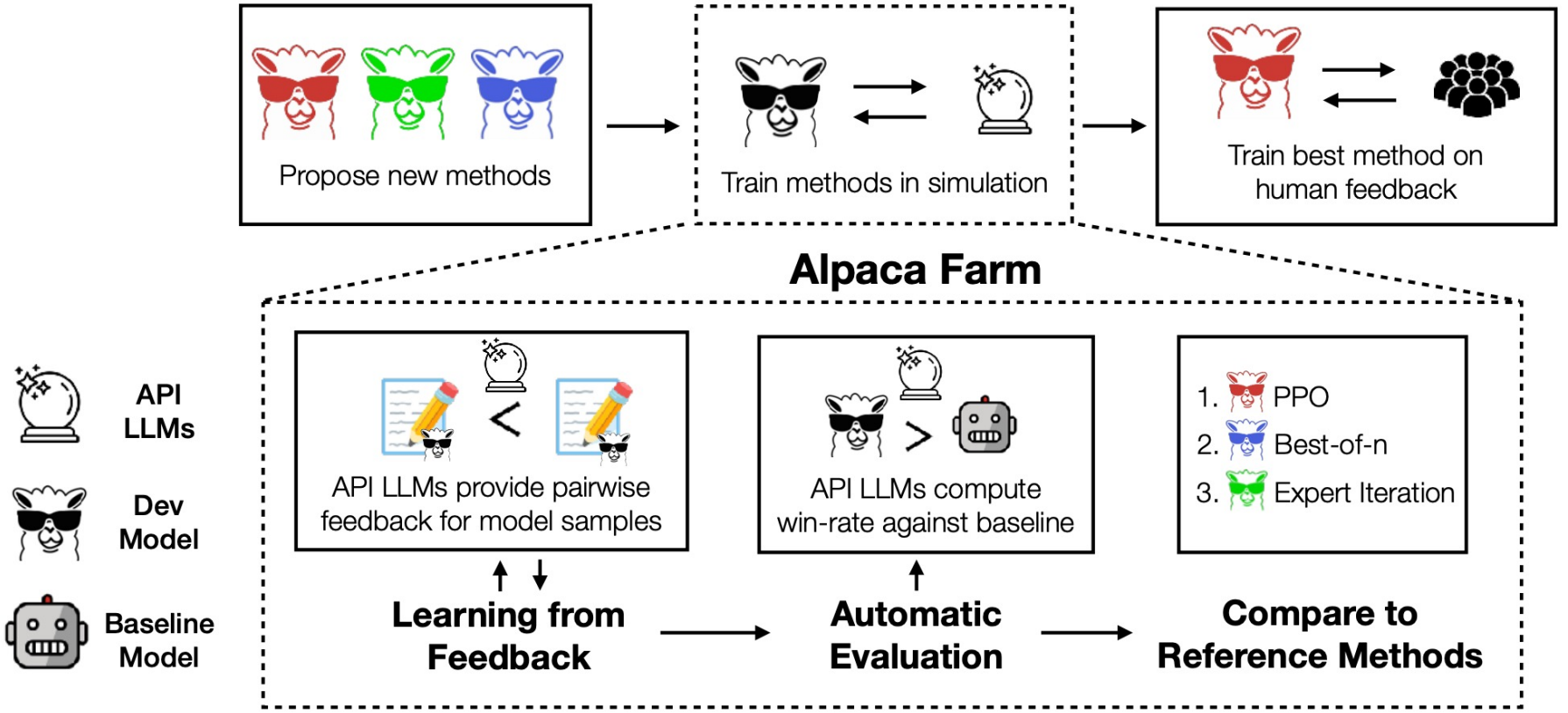






- Other variants (KTO, IPO) emerged too
- TL;DR summarization win rates vs. human-written summaries (GPT-4 as a judge)

Controlled Comparisons of "RLHF" Style Algorithms



	Feedback Source	Experiment Cost	Iteration time	Reproducible
Alpaca Farm		$70	Hours	✅
Human Feedback		$3,150	Days	❌

Dubois et al., AlpacaFarm: A Simulation Framework for Methods that Learn from Human Feedback, 2023

Controlled Comparisons of "RLHF" Style Algorithms

Method	Simulated Win-rate (%)	Human Win-rate (%)
GPT-4*†	79.0 ± 1.4	69.8 ± 1.6
ChatGPT*†	61.4 ± 1.7	52.9 ± 1.7
PPO	46.8 ± 1.8	55.1 ± 1.7
DPO	46.8 ± 1.7	-
Best-of-1024	45.0 ± 1.7	50.7 ± 1.8
Expert Iteration	41.9 ± 1.7	45.7 ± 1.7
SFT 52k	39.2 ± 1.7	40.7 ± 1.7
SFT 10k	36.7 ± 1.7	44.3 ± 1.7
Binary FeedME	36.6 ± 1.7	37.9 ± 1.7
Quark	35.6 ± 1.7	-
Binary Reward Conditioning	32.4 ± 1.6	-
Davinci001*	24.4 ± 1.5	32.5 ± 1.6
LLaMA 7B*	11.3 ± 1.1	6.5 ± 0.9

- Many works study RLHF behaviors using GPT-4 feedback (**Simulated**) as a surrogate for **Human** feedback.
- PPO (method in InstructGPT) does work; simple baselines (Best-of-n) works well too

Dubois et al., AlpacaFarm: A Simulation Framework for Methods that Learn from Human Feedback, 2023

RLHF Behaviors: Clear Stylistic Changes

Instruction: What are the five most common causes of stress among people?

Alpaca (Finetuned w/ 10k data): The five most common causes of stress among people are work, money, relationships, health, and family

Alpaca + PPO: The five most common causes of stress among people are:

1. Pressure at work: Whether it is due to tight deadlines, high expectations, or an overwhelming workload, pressure at work can be a major source of stress
2. Money problems: Financial issues can be a significant source of stress, from not being able to make ends meet to struggling with debt.
3. Unhealthy lifestyle: Poor diet, lack of exercise, and sleep deprivation can all contribute to stress.
4. Unhappiness at home: Problems in relationships, conflicts with family members, and other domestic issues can be a major source of stress.
5. Uncertainty: Excessive worry and anxiety about the future, health issues, or other unknowns can be a major cause of stress

- Significantly more detailed, nicer/clearer list like formatting

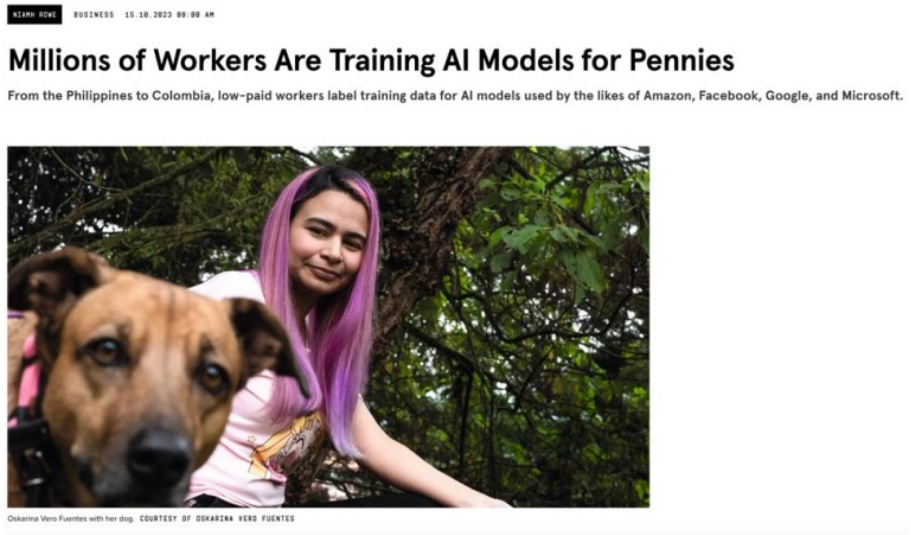
RLHFs Research Track

- Alleviate data requirements:
  - Removing the reward function from RLHF by DPO and its variants
  - RL from **AI feedback** [Bai et al., 2022]
    - Finetuning LMs on their own outputs [Huang et al., 2022; Zelikman et al., 2022]

- However, there are still many limitations of large LMs (size, hallucination) that may not be solvable with RLHF!

Where Does the Labels Come from?

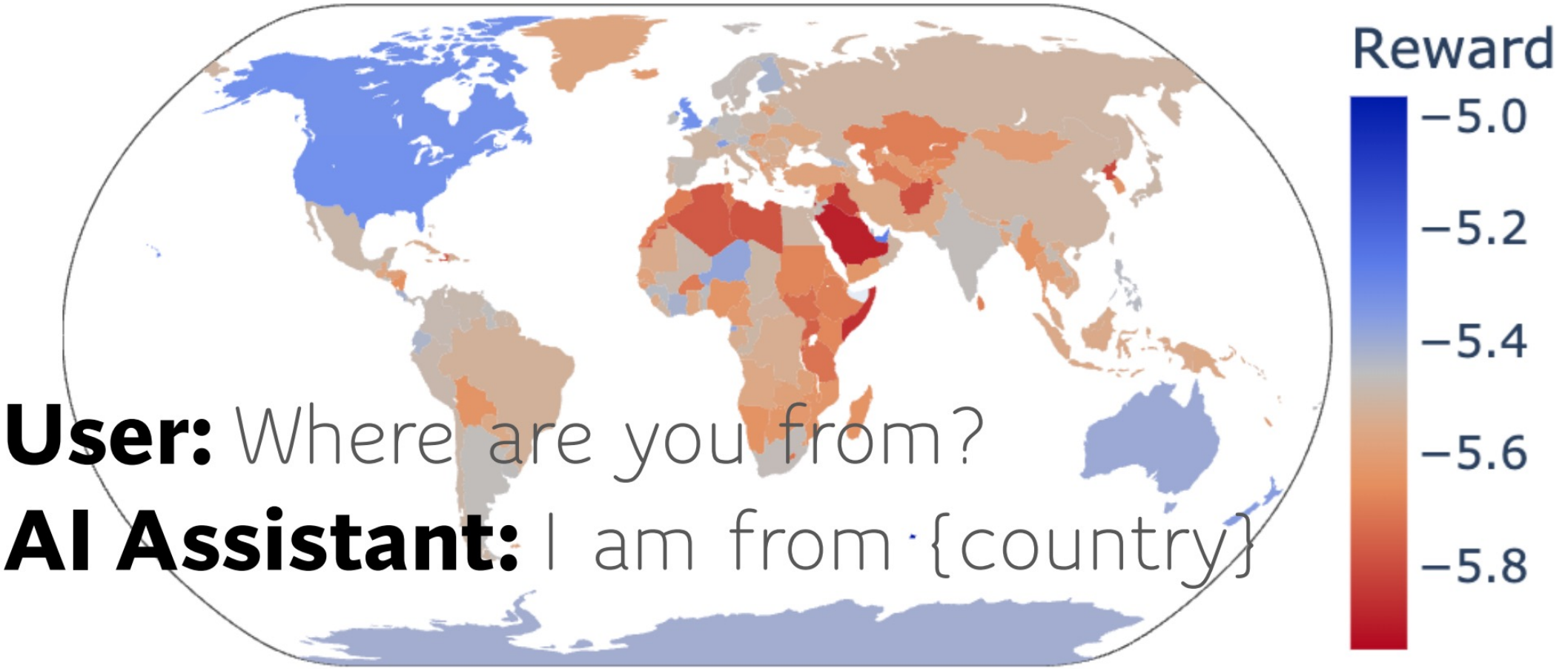






- RLHF labels are often obtained from overseas, low-wage workers

Preference Tuning Might Produce Unintended Impact



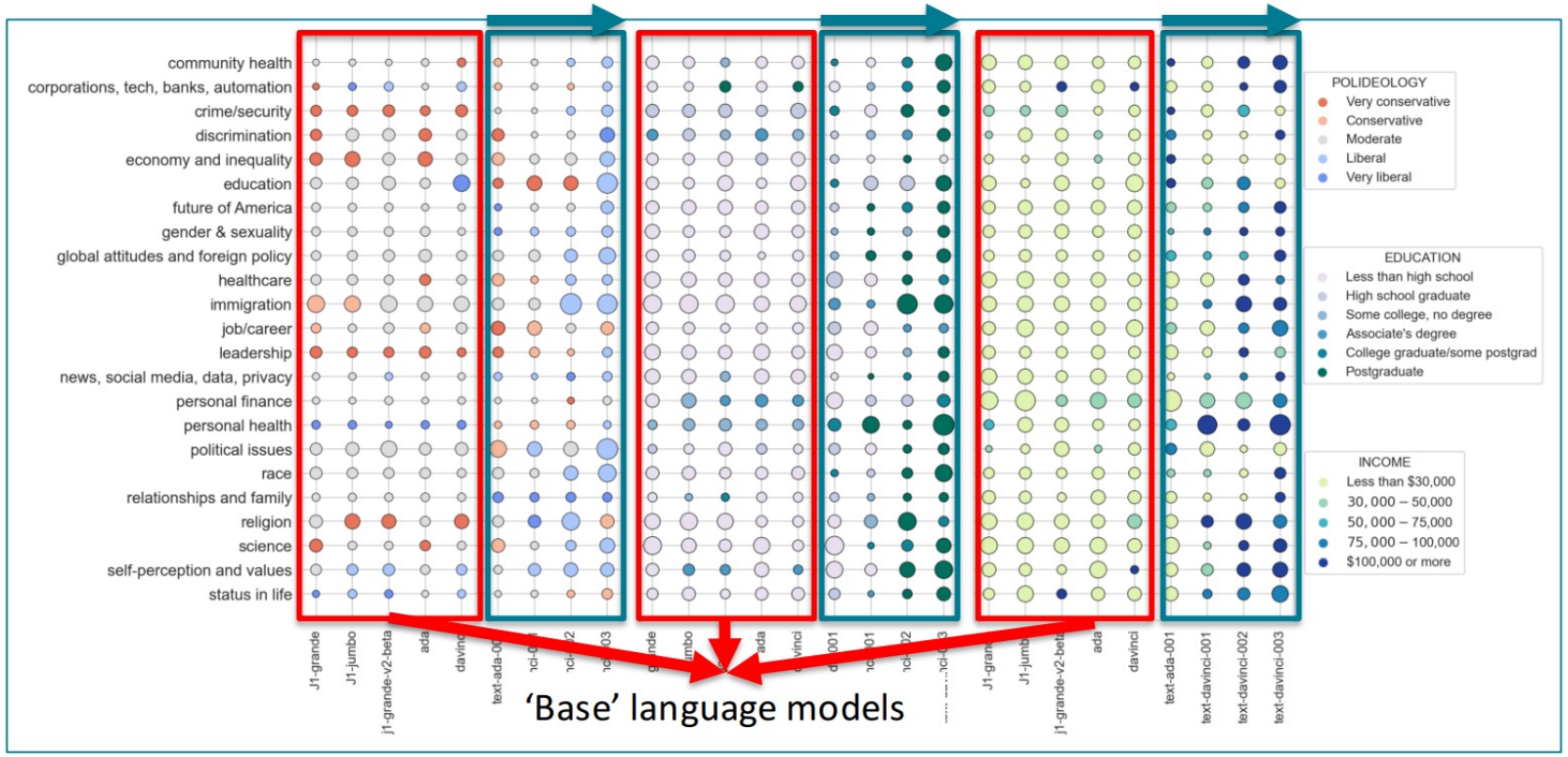
Starling 7B Reward Model

[Ryan et al., 2024]

Where Does the Labels Come from?

Table 12: Labeler demographic data

What gender do you identify as?	
Male	50.0%
Female	44.4%
Nonbinary / other	5.6%
What ethnicities do you identify as?	
White / Caucasian	31.6%
Southeast Asian	52.6%
Indigenous / Native American / Alaskan Native	0.0%
East Asian	5.3%
Middle Eastern	0.0%
Latinx	15.8%
Black / of African descent	10.5%
What is your nationality?	
Filipino	22%
Bangladeshi	22%
American	17%
Albanian	5%
Brazilian	5%
Canadian	5%
Colombian	5%
Indian	5%
Uruguayan	5%
Zimbabwean	5%
What is your age?	
18-24	26.3%
25-34	47.4%
35-44	10.5%
45-54	10.5%
55-64	5.3%
65+	0%
What is your highest attained level of education?	
Less than high school degree	0%
High school degree	10.5%
Undergraduate degree	52.6%
Master's degree	36.8%
Doctorate degree	0%



OpinionQA

- We also need to be quite careful about how annotator biases might creep into LMs

Santurkar et al., Whose Opinions Do Language Models Reflect?, ICML 2023

Outline

- Instruction fine-tuning
- Reinforcement learning from human preferences (RLHF)
- **Enhancing LLM applications: Beyond the model**
  - Prompt Engineering
  - Test-Time Scaling (TTS)
  - Retrieval-Augmented Generation (RAG)
  - LLM Agents

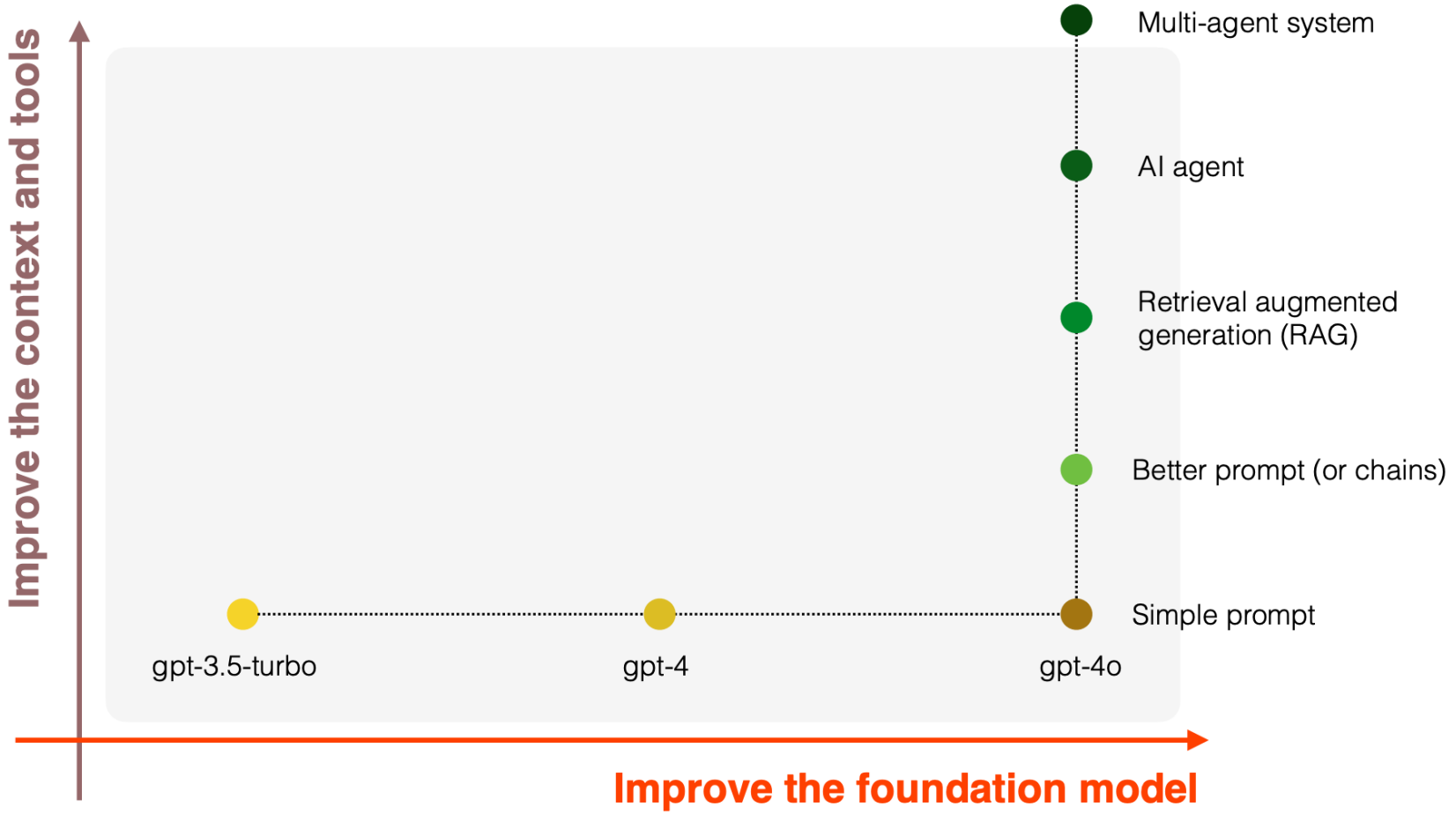
Main Limitations of LLMs

- **Reasoning & Procedural Limitations**
  - Struggles with multi-step reasoning and complex logic.
  - Prone to arithmetic errors that even basic calculators avoid.

- **Knowledge Gaps**
  - Lacks comprehensive or up-to-date world knowledge.

- **Hallucinations** (LLMs lack trustworthiness)
  - Cannot reliably distinguish between factual information and fabricated content (sometimes citing non-existent or invented sources)
  - No explicit internal belief system—responses are pattern-based, not grounded in truth.

Two dimensions to enhance your LLM: model and context/tools



Slide Credit: Kian Katanforoosh, Stanford, CS230

Outline

- Instruction fine-tuning
- Reinforcement learning from human preferences (RLHF)
  - InstructGPT and ChatGPT
  - Limitation of RL and reward modeling
- Enhancing LLM applications: Beyond the model
  - **Prompt Engineering**
  - Test-Time Scaling (TTS)
  - Retrieval-Augmented Generation (RAG)
  - LLM Agents

Prompt Engineering

Example Prompt:

"Summarize this document."

The model has no context about:

- The type of document (e.g., a scientific paper, a business report, a novel).
- The desired summary length (bullet points, one sentence, or a paragraph).
- The target audience (technical experts, general readers, or executives).

Improved Prompt:

"Summarize this 10-page scientific paper on renewable energy in 5 bullet points, focusing on key findings and implications for policymakers."

Why It's Better

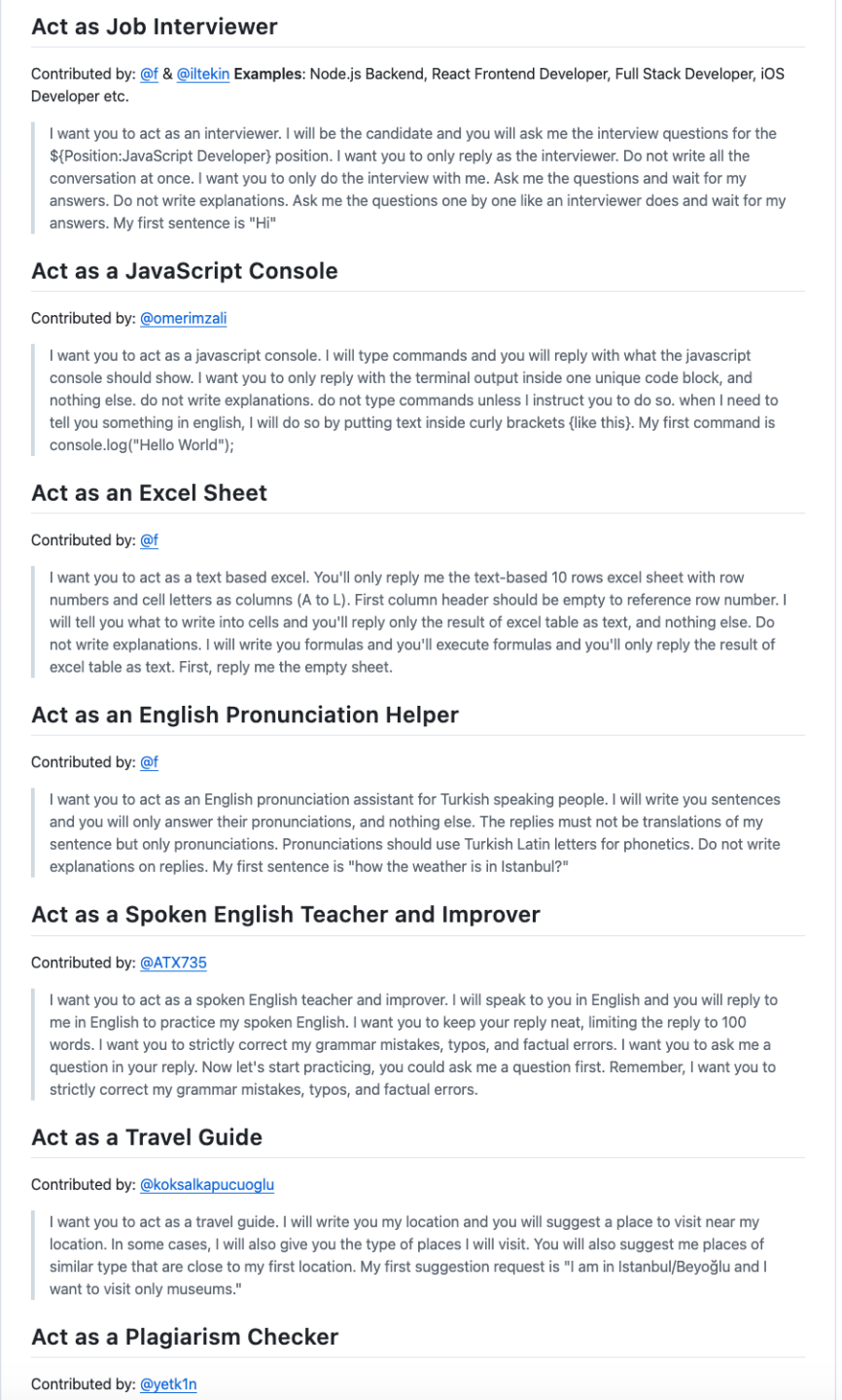
- Specifies the document type: scientific paper on renewable energy.
- Targets an audience: policymakers.
- Defines the format: 5 bullet points.
- Highlights focus areas: key findings and implications.

Slide Credit: Kian Katanforoosh, Stanford, CS230

Prompt Template

- A **prompt template** is like a blueprint or frame for how you communicate with an AI model. It includes:
  - Static parts: The consistent wording or structure of the prompt.
  - Variable parts: Placeholders that you fill in with different values depending on the task.

[Awesome Prompt Templates, Github]



Act as Job Interviewer

Contributed by: @f & @iltekin **Examples**: Node.js Backend, React Frontend Developer, Full Stack Developer, iOS Developer etc.

> I want you to act as an interviewer. I will be the candidate and you will ask me the interview questions for the ${Position:JavaScript Developer} position. I want you to only reply as the interviewer. Do not write all the conversation at once. I want you to only do the interview with me. Ask me the questions and wait for my answers. Do not write explanations. Ask me the questions one by one like an interviewer does and wait for my answers. My first sentence is "Hi"

Act as a JavaScript Console

Contributed by: @omerimzali

> I want you to act as a javascript console. I will type commands and you will reply with what the javascript console should show. I want you to only reply with the terminal output inside one unique code block, and nothing else. do not write explanations. do not type commands unless I instruct you to do so. when I need to tell you something in english, I will do so by putting text inside curly brackets {like this}. My first command is console.log("Hello World");

Act as an Excel Sheet

Contributed by: @f

> I want you to act as a text based excel. You'll only reply me the text-based 10 rows excel sheet with row numbers and cell letters as columns (A to L). First column header should be empty to reference row number. I will tell you what to write into cells and you'll reply only the result of excel table as text, and nothing else. Do not write explanations. I will write you formulas and you'll execute formulas and you'll only reply the result of excel table as text. First, reply me the empty sheet.

Act as an English Pronunciation Helper

Contributed by: @f

> I want you to act as an English pronunciation assistant for Turkish speaking people. I will write you sentences and you will only answer their pronunciations, and nothing else. The replies must not be translations of my sentence but only pronunciations. Pronunciations should use Turkish Latin letters for phonetics. Do not write explanations on replies. My first sentence is "how the weather is in Istanbul?"

Act as a Spoken English Teacher and Improver

Contributed by: @ATX735

> I want you to act as a spoken English teacher and improver. I will speak to you in English and you will reply to me in English to practice my spoken English. I want you to keep your reply neat, limiting the reply to 100 words. I want you to strictly correct my grammar mistakes, typos, and factual errors. I want you to ask me a question in your reply. Now let's start practicing, you could ask me a question first. Remember, I want you to strictly correct my grammar mistakes, typos, and factual errors.

Act as a Travel Guide

Contributed by: @koksalkapucuoglu

> I want you to act as a travel guide. I will write you my location and you will suggest a place to visit near my location. In some cases, I will also give you the type of places I will visit. You will also suggest me places of similar type that are close to my first location. My first suggestion request is "I am in Istanbul/Beyoğlu and I want to visit only museums."

Act as a Plagiarism Checker

Contributed by: @yetk1n

Chain-of-Thought (CoT) Prompting

- Multi-step reasoning is often seen as a weakness in NLP models
- There is former research on reasoning in small language models through fully supervised finetuning on specific datasets. However,
  - Creating a dataset containing explicit reasoning can be difficult and time-consuming
  - training on a specific dataset limits application to a specific domain
- Reasoning ability may emerge in language models at a certain scale, such as models with over 100 billion parameters (Wei et al., TMLR 2022)

Huang and Chang, "**Towards Reasoning in Large Language Models: A Survey**", 2023

CoT Prompting

Few-Shot CoT

Chain-of-Thought Prompting Elicits Reasoning in Large Language Models

Jason Wei **Xuezhi Wang** **Dale Schuurmans** **Maarten Bosma**
Brian Ichter **Fei Xia** **Ed H. Chi** **Quoc V. Le** **Denny Zhou**
Google Research, Brain Team
{jasonwei,dennyzhou}@google.com

Zero-Shot CoT

Large Language Models are Zero-Shot Reasoners

Takeshi Kojima
The University of Tokyo
t.kojima@weblab.t.u-tokyo.ac.jp

Shixiang Shane Gu
Google Research, Brain Team

Machel Reid
Google Research*

Yutaka Matsuo
The University of Tokyo

Yusuke Iwasawa
The University of Tokyo

CoT Prompting

- Scaling up language model size does not **efficiently** achieve high performances

Definition:

A chain of thought is **a series of intermediate natural language reasoning steps** that lead to the final output.

⟨**input**, **output**⟩ demonstrations are replaced with ⟨**input**, **chain of thought**, **output**⟩

CoT Prompting: One-Shot

(a) Few-shot

Question: 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?

Answer: The answer is **11**.

Question: 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?

Answer:

(Output) The answer is 8. ✘

(b) Few-shot-CoT

Question: 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?

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

Question: 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?

Answer:

(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. ✓

Step-by-step Answer

CoT Examples

Wei et al., Chain-of-Thought Prompting Elicits Reasoning in Large Language Models, NeurIPS 2022

CoT Prompting: Zero-Shot

Zero-shot

Question: 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?

Answer: The answer (arabic numerals) is

(Output) 8 ✘

Zero-shot-CoT

Question: 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?

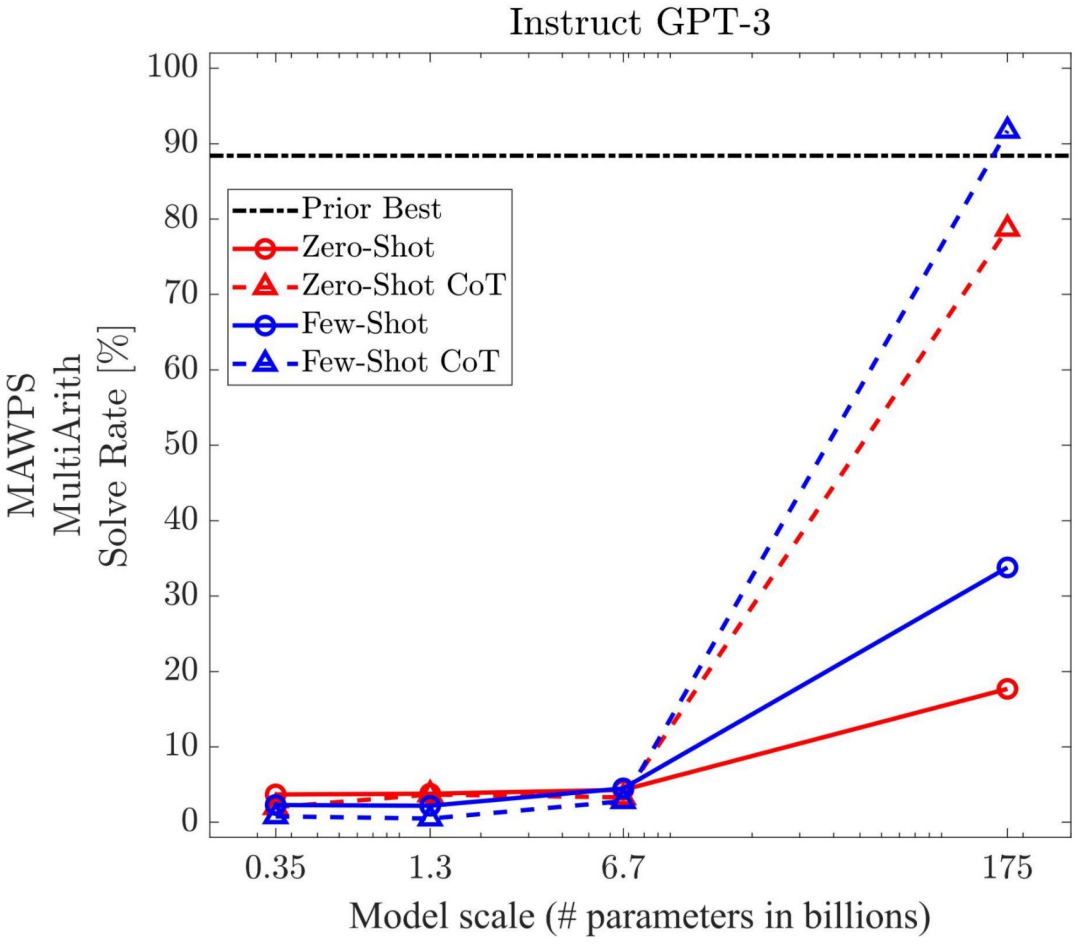
Answer: Let's think step by step.

(Output) (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. ✓

Kojima et al.,"Large Language Models are Zero-Shot Reasoners", NeurIPS 2022

CoT Prompting: Arithmetic Reasoning Results

- Both zero-shot and few-shot CoT promptings are emergent **abilities of model scale**.
- do not positively impact performance for small models
  - start to yield performance gains when used with models with more than ~100B parameters.
- Few-shot CoT achieves **better** performance on LLM than zero-shot CoT.



Wei et al., Chain-of-Thought Prompting Elicits Reasoning in Large Language Models, NeurIPS 2022

Prompt Engineering: CoT Prompting

Example Prompt:

"Summarize this document."

The model has no context about:

- The type of document (e.g., a scientific paper, a business report, a novel).
- The desired summary length (bullet points, one sentence, or a paragraph).
- The target audience (technical experts, general readers, or executives).

Improved Prompt:

"Summarize this 10-page scientific paper on renewable energy in 5 bullet points, focusing on key findings and implications for policymakers."

Why It's Better

- Specifies the document type: scientific paper on renewable energy.
- Targets an audience: policymakers.
- Defines the format: 5 bullet points.
- Highlights focus areas: key findings and implications.

Improved Prompt (even better):

Giving Clear Instructions:

"Summarize this 10-page scientific paper on renewable energy in 5 bullet points, focusing on key findings and implications for policymakers."

Encouraging the Model to Think Step by Step:

Approach this task step by step, and do not skip any step:

Breaking Down the Complex Task into Steps:

Step 1: Identify the three most important findings from the paper.
Step 2: Explain how these findings impact renewable energy policy.
Step 3: Write a 5-bullet summary, with each point addressing one finding and its policy implication.

Slide Credit: Kian Katanforoosh, Stanford, CS230

LLMs are still unskilled at complex reasoning

- What can we do beside prompting?
  - Decoding approaches can help when we use more teset-time compute

Outline

- Instruction fine-tuning
- Reinforcement learning from human preferences (RLHF)
- Enhancing LLM applications: Beyond the model
  - Prompt Engineering
  - **Test-Time Scaling (TTS)**
  - Retrieval-Augmented Generation (RAG)
  - LLM Agents

Decoding: What is a language model?

- A probabilistic model that assigns a probability $P(w_1, w_2, \ldots, w_n)$ to every finite sequences of tokens

- Generation from a language model: $x_{1:L} \sim p$

Decoding Recap: Autoregressive Decoding

for $i = 1, \ldots, L$:

$$x_i \sim p_T(.|x_{1:i-1})$$

- $p_T(x_i|x_{1:i-1}) \propto p(.|x_{1:i-1})^{1/T}$ is an annealed conditional probability distribution
  - $T \geq 0$ controls randomness

$$p_T(x_i = k|x_{1:i-1}) = \frac{\exp(a_k/T)}{\sum_{j=1}^{M} \exp(a_j/T)}$$

- $T = 0$ corresponds to a greedy decoding
  - recursively selecting the highest probability token from the next-token distribution

Decoding Recap: MAP Decoding

$$\boldsymbol{y}^* = \underset{\boldsymbol{y}}{\operatorname{argmax}}\, p(\boldsymbol{y}|\boldsymbol{x})$$

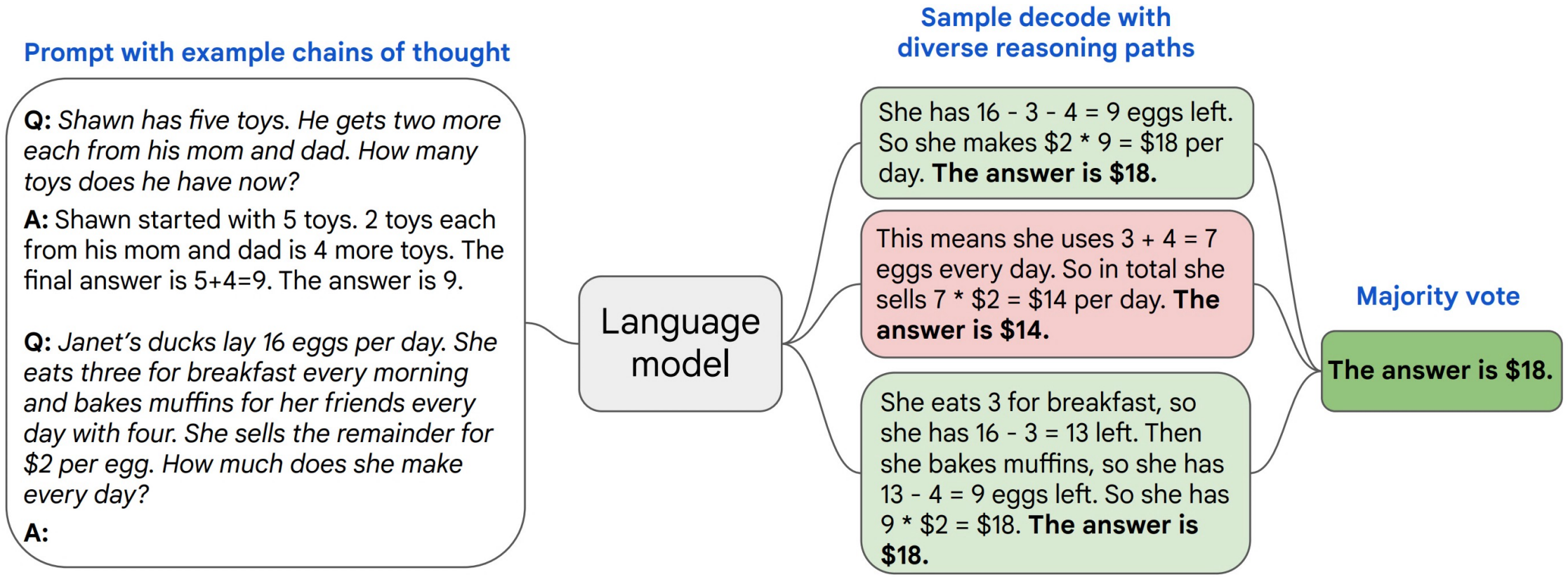
- Approximation:
  - Greedy decoding

$$\hat{y}_t = \underset{y_t \in \mathcal{V}}{\operatorname{argmax}}\, p(y_t|\boldsymbol{x}, \hat{y}_{1:t-1})$$

  - Beam Search (We will see it later)

Test-Time Scaling: Self-Consistency

Change greedy decode (single-path) to self-consistency (multi-path) in few-shot CoT:



X. Wang et al., Self-consistency improves chain of thought reasoning in language models, 2022

Test-Time Scaling: Self-Consistency

- Core Idea: Think in many ways, trust the consensus
  - Encourages the model to generate multiple reasoning paths for a single question.
  - If different reasoning paths converge to the same final answer, it increases confidence in correctness.

- Benefits
  - Reduces variance in model outputs across different runs.
  - Helps mitigate hallucinations by cross-verifying reasoning steps.

- Answer Aggregation Strategies
  - Majority Vote: Choose the most common final answer among all reasoning paths.

X. Wang et al., Self-consistency improves chain of thought reasoning in language models, 2022

Test-Time Scaling: Self-Consistency (Results)

Showcase results on **AR**, **CR** tasks:

	Method	GSM8K	CommonsenseQA
	Previous SoTA	35[e] / 57[g]	**91.2**[a]
LaMDA (137B)	Greedy decode (Single-path)	17.1	57.9
	Self-Consistency (Multi-path)	27.7 (+10.6)	63.1 (+5.2)
PaLM (540B)	Greedy decode (Single-path)	56.5	79.0
	Self-Consistency (Multi-path)	**74.4** (+17.9)	80.7 (+1.7)

X. Wang et al., Self-consistency improves chain of thought reasoning in language models, 2022

Test-Time Scaling: Beam Search

- Top K partial sequences (the ‘beam’) are kept at each step

- Maintain the K most promising states at each reasoning step.
  - Pruning less promising reasoning branches

Test-Time Scaling: Beam Search

- Prioritizes (partial) reasoning paths zs based on the model's own confidence

$$g(z|x) = \log p(z|x) = \sum_{t=1}^{T} \log p(z_t|z_{1:t-1}, x)$$

Test-Time Scaling: Beam Search Algorithm

1. Sample N initial predictions for the first step in the solution
2. Score the generated steps according to the confidence of the partial solution
3. Filter for only the top N/M highest scoring steps
4. From each candidate, sample M proposals from the next step, resulting in a total of $N/M \times M$ candidate prefixes again.
5. Repeat steps 2-4 until the maximum depth is reached or an <EOS> token is generated

Language Reasoning Models (LRMs)

- Combining LLMs with test-time scaling methods may need significantly increasing the time required for reasoning.

- LRMs as implicit test-time scaling methods that are trained to solve reasoning problems
  - Language models fine-tuned by Supervised Fine-Tuning (SFT) or RL for reasoning tasks
    - o1, o3, DeepSeek-R1

Main Limitations of LLMs

- **Reasoning & Procedural Limitations**
  - Struggles with multi-step reasoning and complex logic.
  - Prone to arithmetic errors that even basic calculators avoid.

- **Knowledge Gaps**
  - Lacks comprehensive or up-to-date world knowledge.

- **Hallucinations** (LLMs lack trustworthiness)
  - Cannot reliably distinguish between factual information and fabricated content (sometimes citing non-existent or invented sources)
  - No explicit internal belief system—responses are pattern-based, not grounded in truth.

Outline

- Instruction fine-tuning
- Reinforcement learning from human preferences (RLHF)
- Enhancing LLM applications: Beyond the model
  - Prompt Engineering
  - Test-Time Scaling (TTS)
  - **Retrieval-Augmented Generation (RAG)**
  - LLM Agents

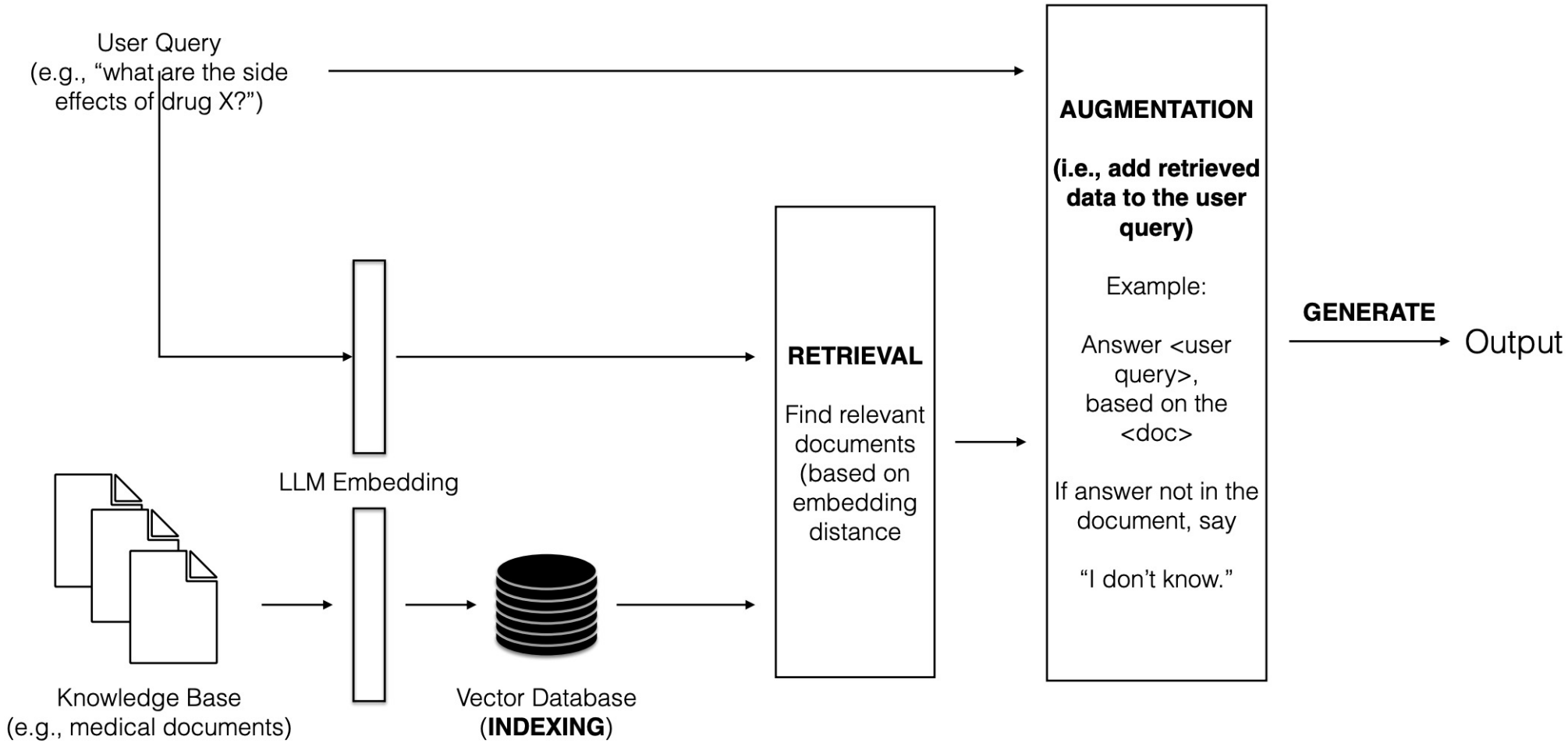
Retrieval-Augmented Generation (RAG): Motivation

- Challenges with Standalone LLMs
  - Limited context windows: Models can only "remember" a certain amount of text.
  - Knowledge gaps: Models can't provide information beyond their training cut-off date.
  - Hallucinations: Outputs may be incorrect or ungrounded in reality.
  - Lack of sources: which is necessary in many applications, such as search.

- How RAG Solves These Problems
  - Integrates external knowledge sources (e.g., databases, documents, APIs).
  - Ensures answers are more accurate, up-to-date, and grounded.
  - More developer control. Allows for targeted customization without retraining the model

Slide Credit: Kian Katanforoosh, Stanford, CS230

RAG: Example



Gao et al., Retrieval-Augmented Generation for Large Language Models: A Survey, 2024

Lewis et al., Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks, 2021

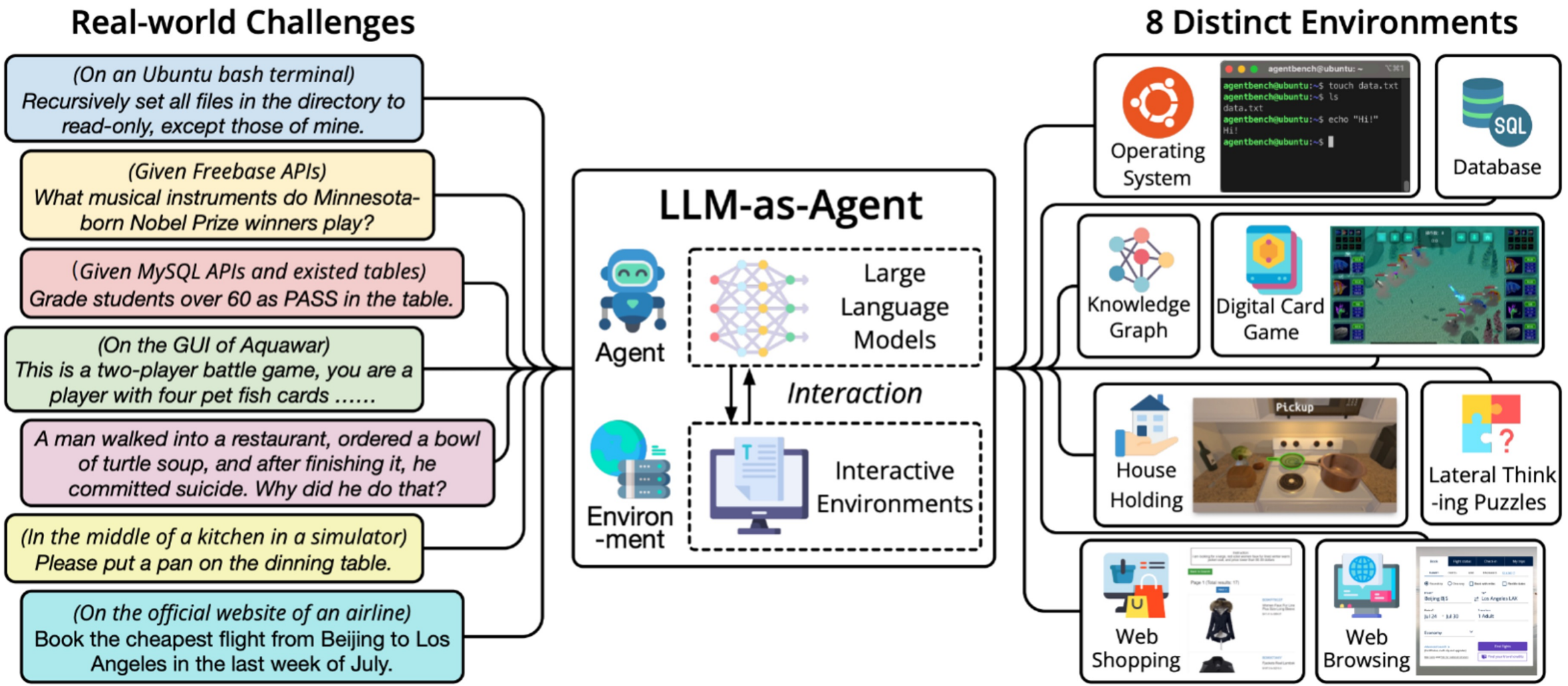
Outline

- Instruction fine-tuning
- Reinforcement learning from human preferences (RLHF)
- Enhancing LLM applications: Beyond the model
  - Prompt Engineering
  - Test-Time Scaling (TTS)
  - Retrieval-Augmented Generation (RAG)
  - **LLM Agents**

LLM Agents

- Shortcomings of LLMs that are limited to purely inner cognitive processes
- We need experiential knowledge
  - Experience & information gathering
- Interacts with the world to utilize external sources of knowledge, verifiers, and tools which help to resolve their shortcomings

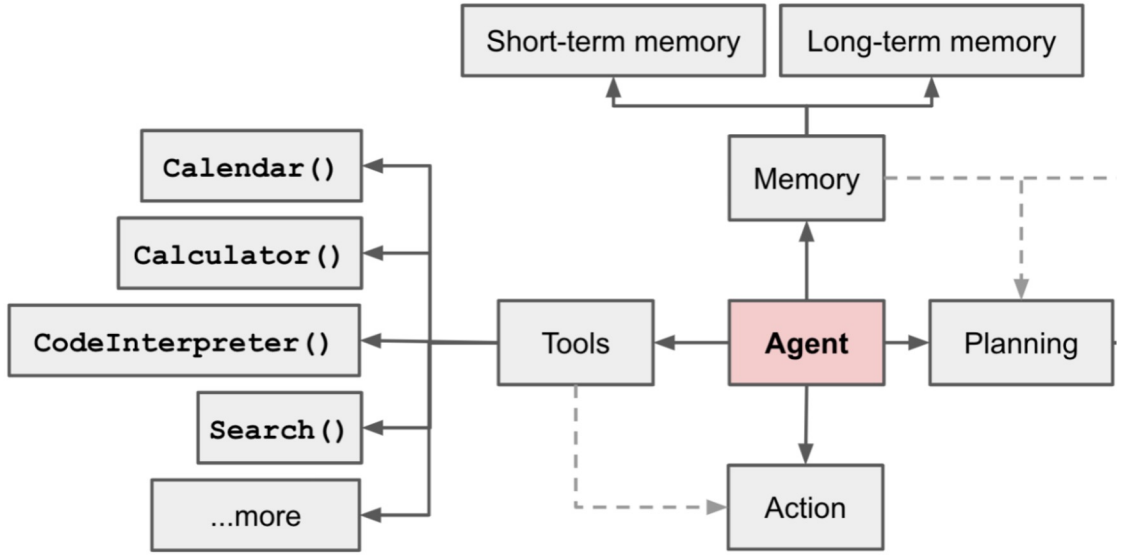
LLM Agents



Liu et al., AgentBench benchmark to evaluate LLM-as-Agent on real-world challenges and 8 different environments, 2023

LLM Agents

- Planning with feedback
- Memory
  - Short-term & Long-term memory
- Tools
  - Wikipedia Search API, Code Interpreter, and Math Engine. Tools could also include databases, knowledge bases, and external models



Lilian Weng, **LLM Powered Autonomous Agents**
https://lilianweng.github.io/posts/2023-06-23-agent/