Posttraining and Alignment:
Instruction Tuning, RLHF, PPO, DPO

CS6120: Natural Language Processing
Northeastern University

David Smith
with slides from Diyi Yang, Jesse Mu, Nathan Lambert, Chris Manning

Predicting language ≠ 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.
```

Language models are not *aligned* with user intent [Ouyang et al., 2022]

Finetuning to the rescue!

Predicting language $\neq$ user intent

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.

Language models are not *aligned* with user intent [Ouyang et al., 2022]

Finetuning to the rescue!

The story so far...

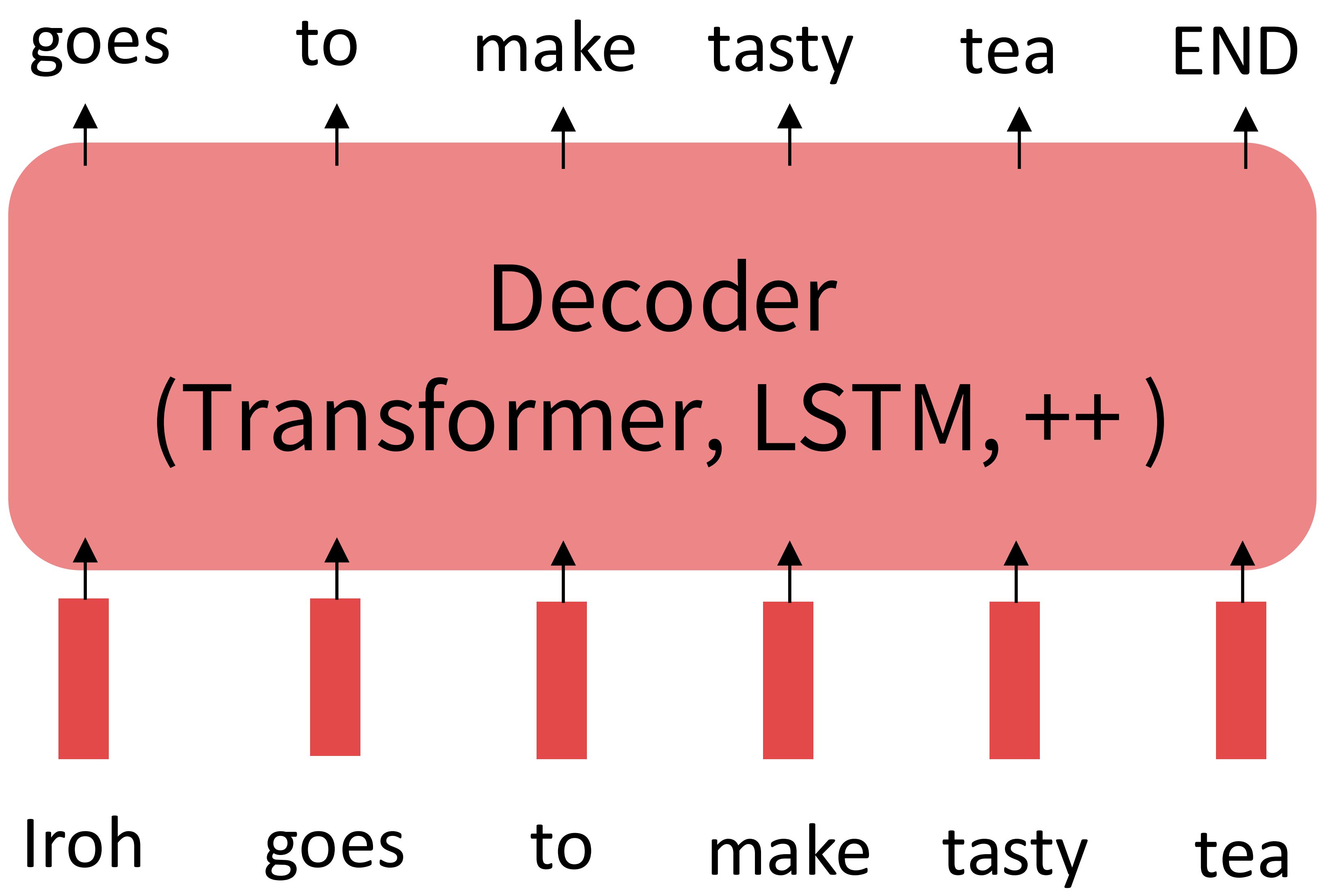
- Train a language model on millions, billions, or trillions of tokens
- Create training objectives from **auxiliary tasks** that we can automatically generate from plain text
  - Teacher-forcing autoregressive prediction, masked language modeling, next sentence prediction, span denoising…
- Collect a small amount of labeled data for a particular task and **fine-tune** (some of) the weights
- But fine tuning still takes a while, and there are a lot of tasks!

The pretraining/finetuning paradigm

Pretraining can improve NLP applications by serving as parameter initialization.

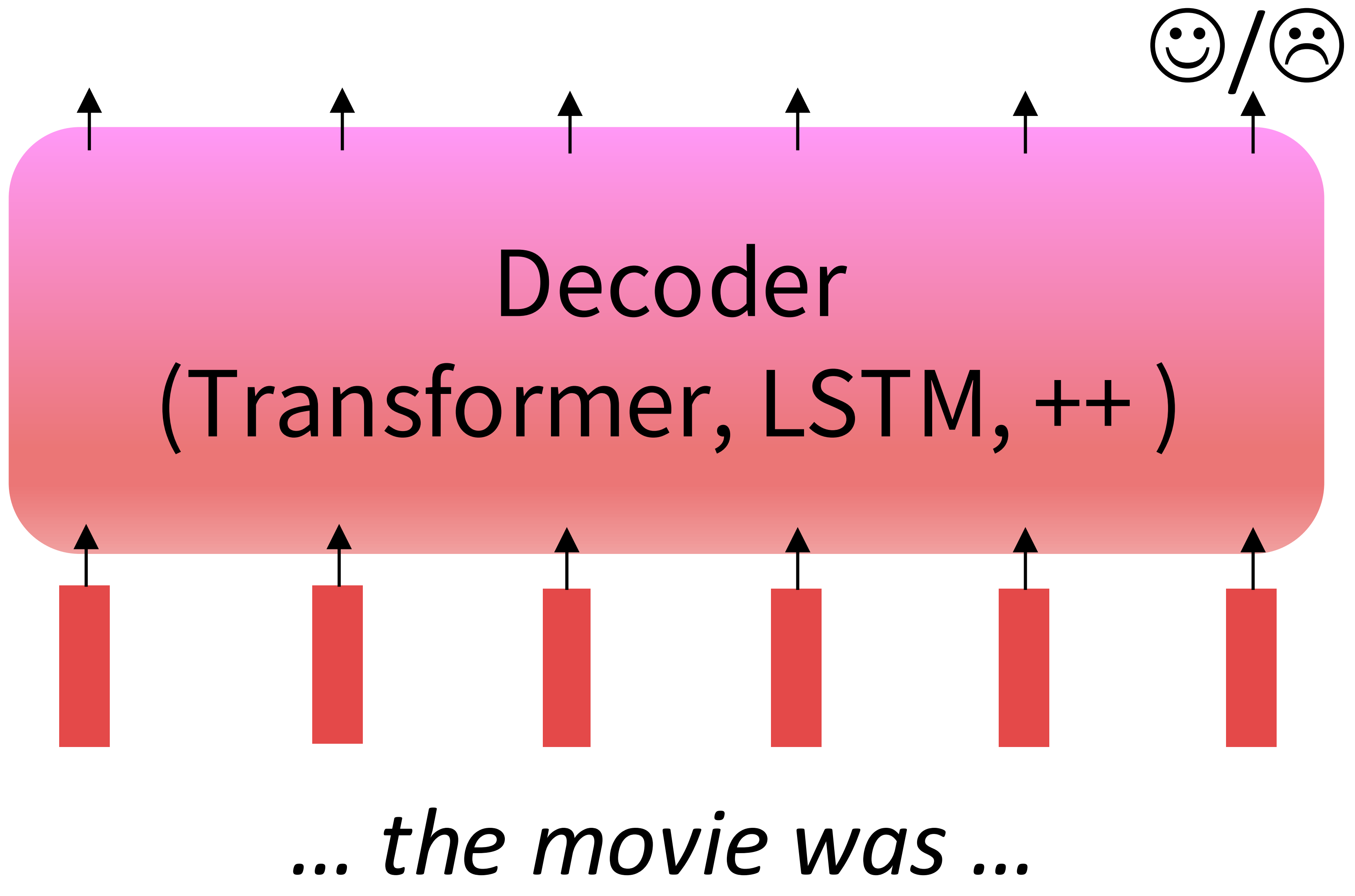
Step 1: Pretrain (on language modeling)

Lots of text; learn general things!



Step 2: Finetune (on your task)

Not many labels; adapt to the task!

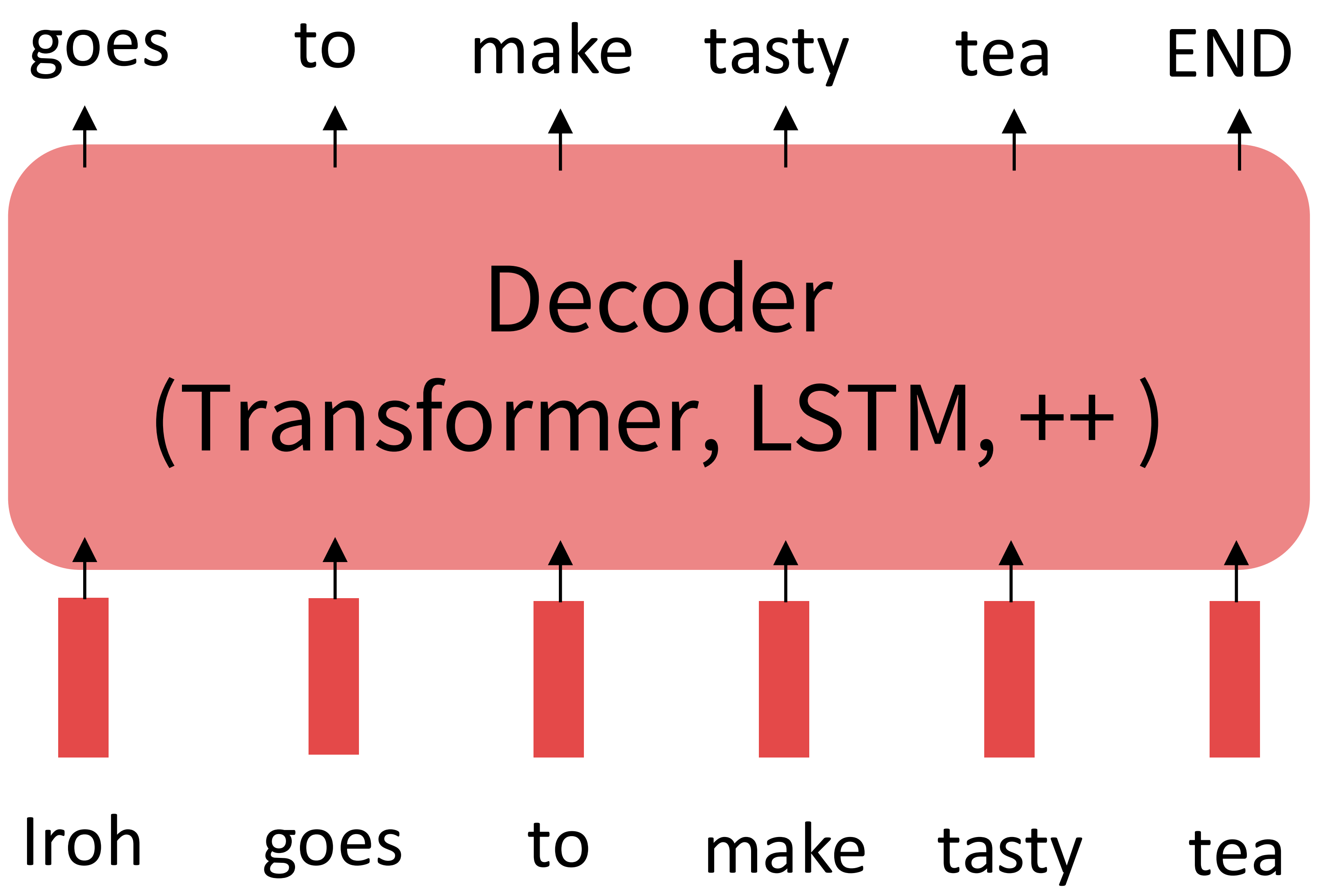


The pretraining/finetuning paradigm

Pretraining can improve NLP applications by serving as parameter initialization.

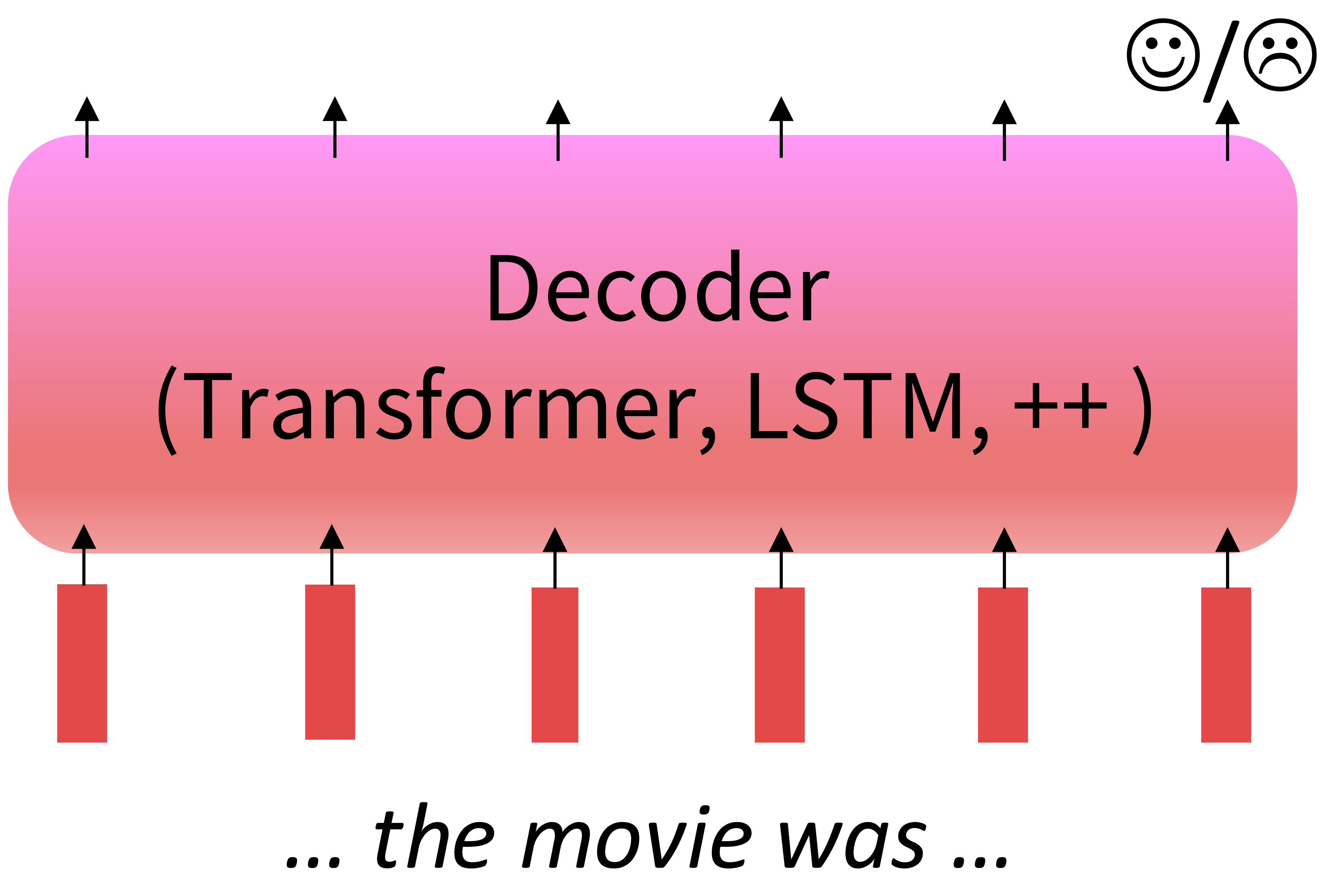
Step 1: Pretrain (on language modeling)

Lots of text; learn general things!



Step 2: Finetune (on many tasks)

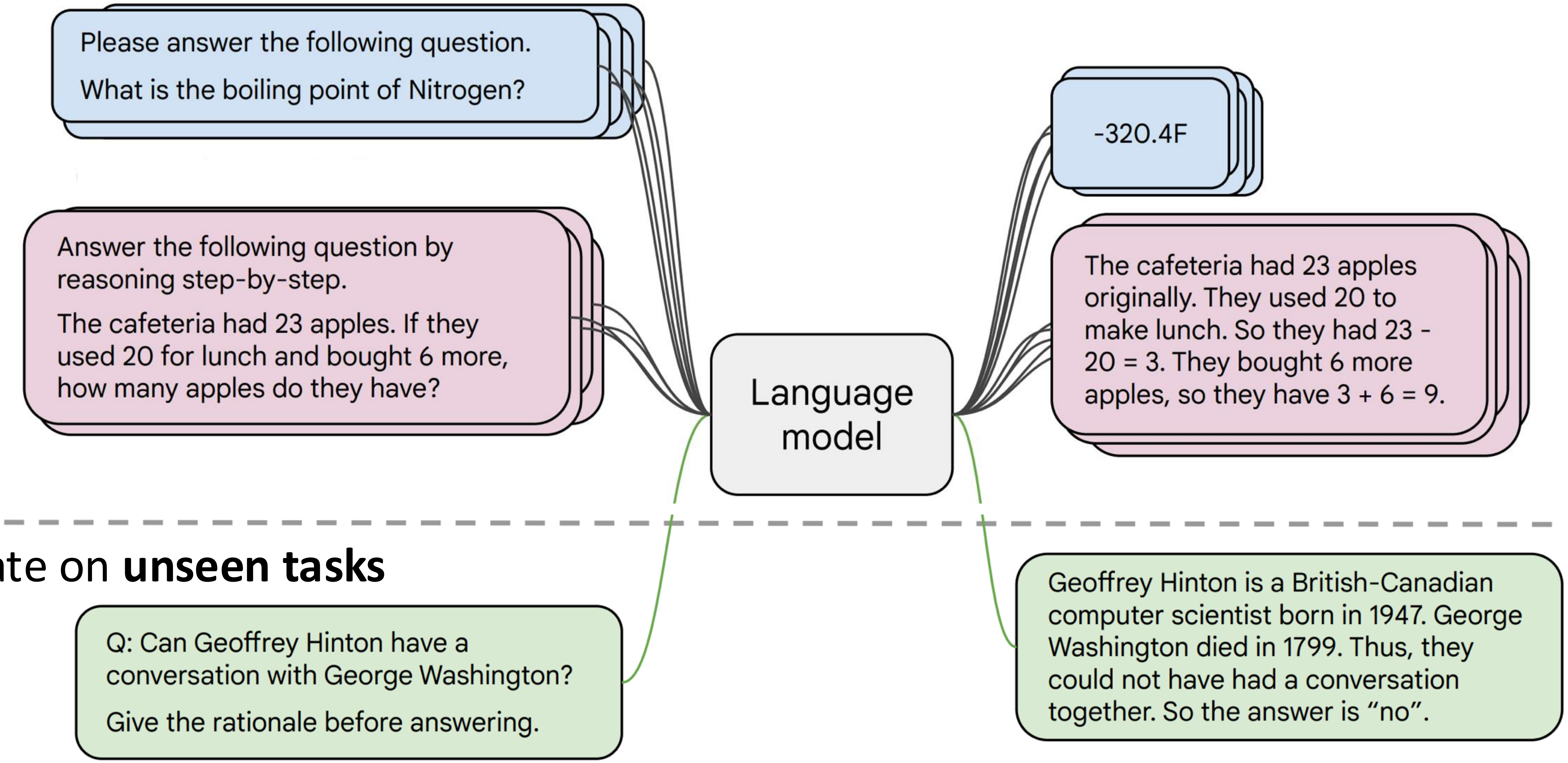
~~Not~~ many labels; adapt to the tasks!



Instruction Tuning

Instruction finetuning

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



- Evaluate on **unseen tasks**

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

Instruction pretaining

- As is usually the case, **data + model scale** is key for this to work!

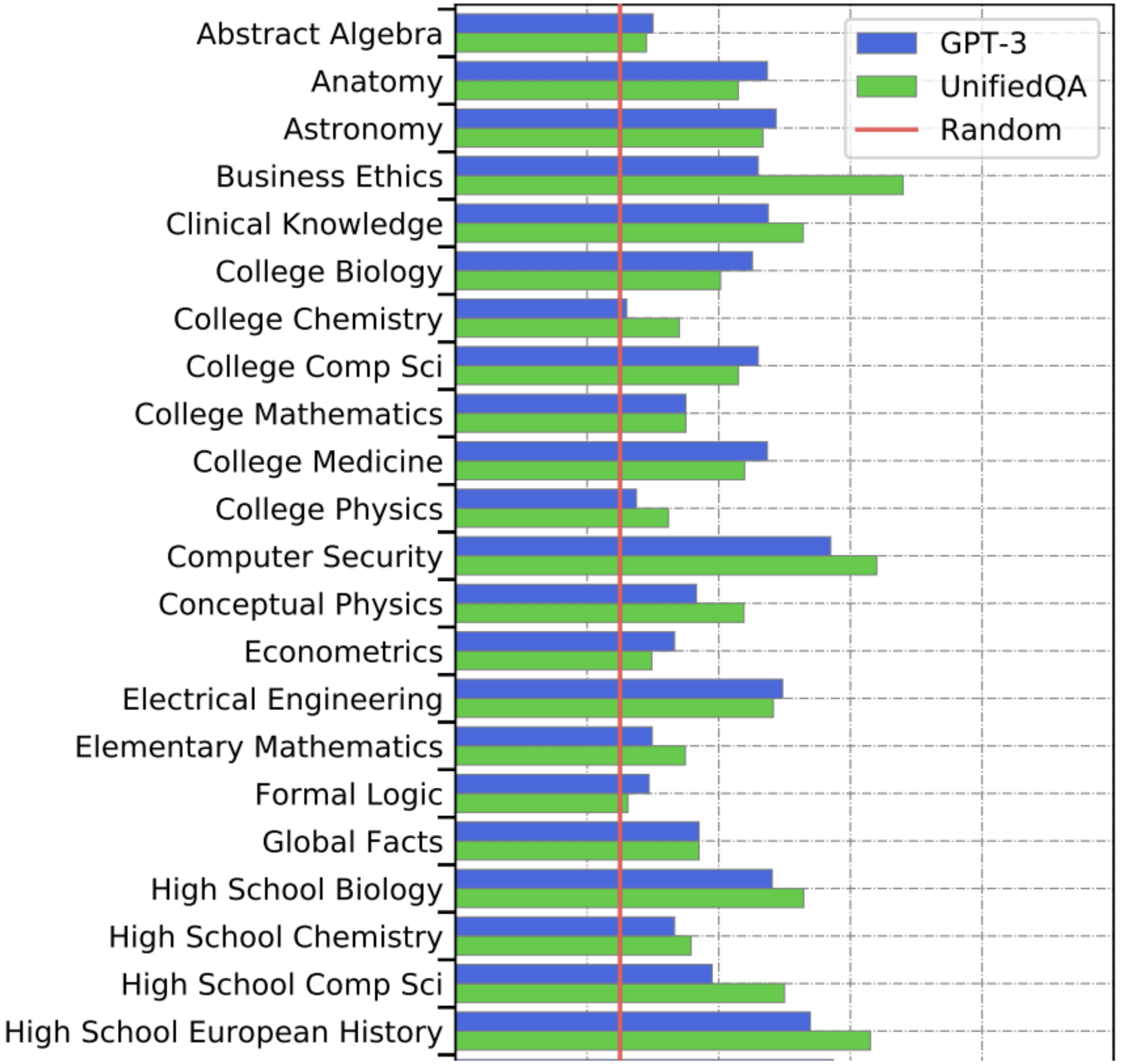
- **Super-NaturalInstructions** dataset contains **over 1.6K tasks, 3M+** examples
  - Classification, sequence tagging, rewriting, translation, QA…

Q: how do we evaluate such a model?

More diverse evaluations

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

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



Examples from MMLU

Astronomy

What is true for a type-Ia supernova?

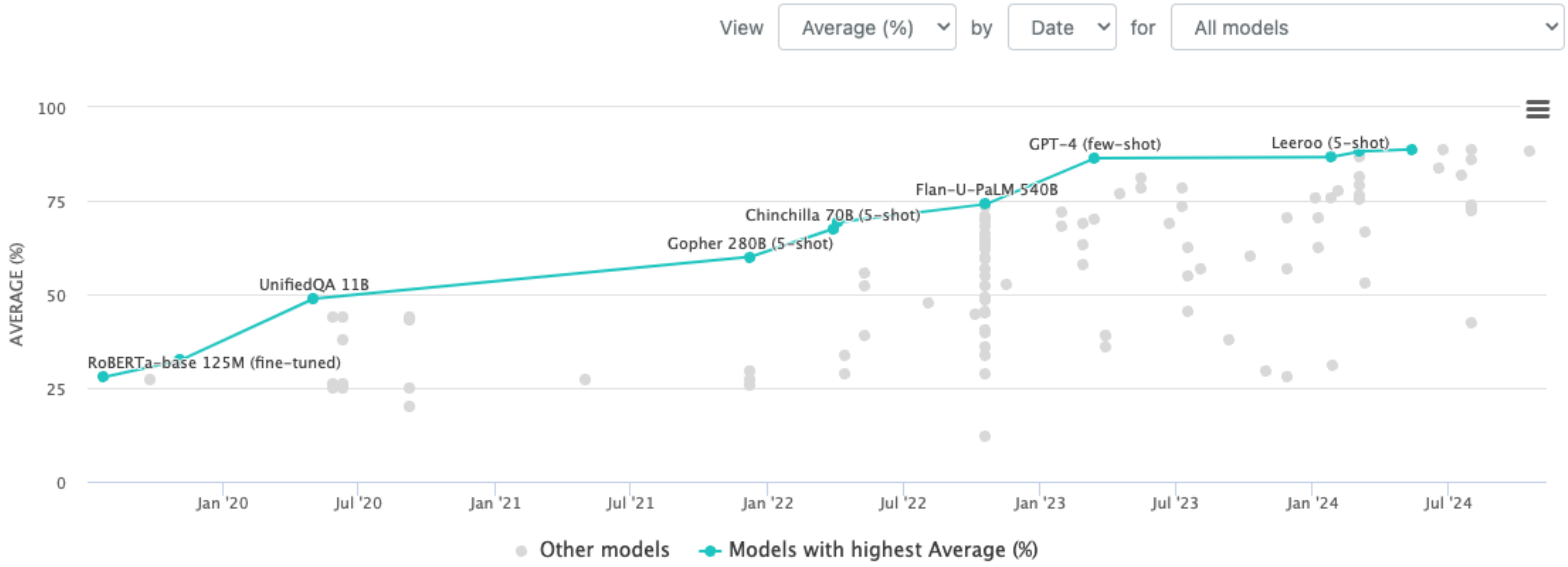
A. This type occurs in binary systems.
B. This type occurs in young galaxies.
C. This type produces gamma-ray bursts.
D. This type produces high amounts of X-rays.

High School Biology

In a population of giraffes, an environmental change occurs that favors individuals that are tallest. As a result, more of the taller individuals are able to obtain nutrients and survive to pass along their genetic information. This is an example of

A. directional selection.
B. stabilizing selection.
C. sexual selection.
D. disruptive selection

Progress on MMLU



- Rapid, impressive progress on challenging knowledge-intensive benchmarks

Even Bigger Evaluations

BIG-Bench [[Srivastava et al., 2022](#)]
200+ tasks, spanning:

https://github.com/google/BIG-bench/blob/main/bigbench/benchmark_tasks/README.md

Beyond the Imitation Game: Quantifying and extrapolating the capabilities of language models

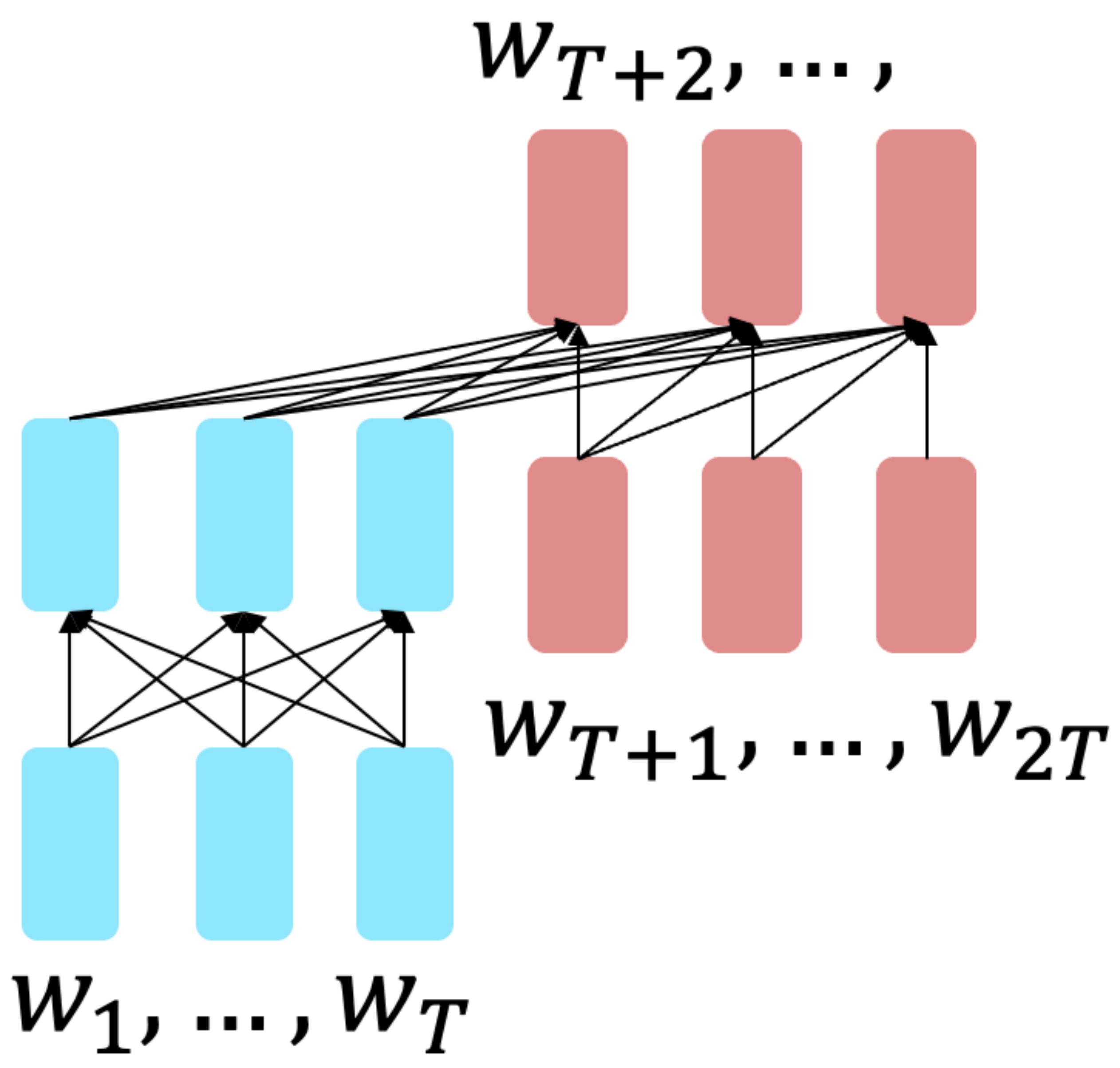
Alphabetic author list:*

Aarohi Srivastava, Abhinav Rastogi, Abhishek Rao, Abu Awal Md Shoeb, Abubakar Abid, Adam Fisch, Adam R. Brown, Adam Santoro, Aditya Gupta, Adrià Garriga-Alonso, Agnieszka Kluska, Aitor Lewkowycz, Akshat Agarwal, Alethea Power, Alex Ray, Alex Warstadt, Alexander W. Kocurek, Ali Safaya, Ali Tazarv, Alice Xiang, Alicia Parrish, Allen Nie, Aman Hussain, Amanda Askell, Amanda Dsouza, Ambrose Slone, Ameet Rahane, Anantharaman S. Iyer, Anders Andreassen, Andrea Madotto, Andrea Santilli, Andreas Stuhlmüller, Andrew Dai, Andrew La, Andrew Lampinen, Andy Zou, Angela Jiang, Angelica Chen, Anh Vuong, Animesh Gupta, Anna Gottardi, Antonio Norelli, Anu Venkatesh, Arash Gholamidavoodi, Arfa Tabassum, Arul Menezes, Arun Kirubarajan, Asher Mullokandov, Ashish Sabharwal, Austin Herrick, Avia Efrat, Aykut Erdem, Ayla Karakaş, B. Ryan Roberts, Bao Sheng Loe, Barret Zoph, Bartłomiej Bojanowski, Batuhan Özyurt, Behnam Hedayatnia, Behnam Neyshabur, Benjamin Inden, Benno Stein, Berk Ekmekci, Bill Yuchen Lin, Blake Howald, Bryan Orinion, Cameron Diao, Cameron Dour, Catherine Stinson, Cedrick Argueta, César Ferri Ramírez, Chandan Singh, Charles Rathkopf, Chenlin Meng, Chitta Baral, Chiyu Wu, Chris Callison-Burch, Chris Waites, Christian Voigt, Christopher D. Manning, Christopher Potts, Cindy Ramirez, Clara E. Rivera, Clemencia Siro, Colin Raffel, Courtney Ashcraft, Cristina Garbacea, Damien Sileo, Dan Garrette, Dan Hendrycks, Dan Kilman, Dan Roth, Daniel Freeman, Daniel Khashabi, Daniel Levy, Daniel Moseguí González, Danielle Perszyk, Danny Hernandez, Danqi Chen, Daphne Ippolito, Dar Gilboa, David Dohan, David Drakard, David Jurgens, Debajyoti Datta, Deep Ganguli, Denis Emelin, Denis Kleyko, Deniz Yuret, Derek Chen, Derek Tam, Dieuwke Hupkes, Diganta Misra, Dilyar Buzan, Dimitri Coelho Mollo, Diyi Yang, Dong-Ho Lee, Dylan Schrader, Ekaterina Shutova, Ekin Dogus Cubuk, Elad Segal, Eleanor Hagerman, Elizabeth Barnes, Elizabeth Donoway, Ellie Pavlick, Emanuele Rodola, Emma Lam, Eric Chu, Eric Tang, Erkut Erdem, Ernie Chang, Ethan A. Chi, Ethan Dyer, Ethan Jerzak, Ethan Kim, Eunice Engefu Manyasi, Evgenii Zheltonozhskii, Fanyue Xia, Fatemeh Siar, Fernando Martínez-Plumed, Francesca Happé, Francois Chollet, Frieda Rong, Gaurav Mishra, Genta Indra Winata, Gerard de Melo, Germán Kruszewski, Giambattista Parascandolo, Giorgio Mariani, Gloria Wang, Gonzalo Jaimovitch-López, Gregor Betz, Guy Gur-Ari, Hana Galijasevic, Hannah Kim, Hannah Rashkin, Hannaneh Hajishirzi, Harsh Mehta, Hayden Bogar, Henry Shevlin, Hinrich Schütze, Hiromu Yakura, Hongming Zhang, Hugh Mee Wong, Ian Ng, Isaac Noble, Jaap Jumelet, Jack Geissinger, Jackson Kernion, Jacob Hilton, Jaehoon Lee, Jaime Fernández Fisac, James B. Simon, James Koppel, James Zheng, James Zou, Jan Kocoń, Jana Thompson, Janelle Wingfield, Jared Kaplan, Jarema Radom, Jascha Sohl-Dickstein, Jason Phang, Jason Wei, Jason Yosinski, Jekaterina Novikova, Jelle Bosscher, Jennifer Marsh, Jeremy Kim, Jeroen Taal, Jesse Engel, Jesujoba Alabi, Jiacheng Xu, Jiaming Song, Jillian Tang, Joan Waweru, John Burden, John Miller, John U. Balis, Jonathan Batchelder, Jonathan Berant, Jörg Frohberg, Jos Rozen, Jose Hernandez-Orallo, Joseph Boudeman, Joseph Guerr, Joseph Jones, Joshua B. Tenenbaum, Joshua S. Rule, Joyce Chua, Kamil Kanclerz, Karen Livescu, Karl Krauth, Karthik Gopalakrishnan, Katerina Ignatyeva, Katja Markert, Kaustubh D. Dhole, Kevin Gimpel, Kevin Omondi, Kory Mathewson, Kristen Chiafullo, Ksenia Shkaruta, Kumar Shridhar, Kyle McDonell, Kyle Richardson, Laria Reynolds, Leo Gao, Li Zhang, Liam Dugan, Lianhui Qin, Lidia Contreras-Ochando, Louis-Philippe Morency, Luca Moschella, Lucas Lam, Lucy Noble, Ludwig Schmidt, Luheng He, Luis Oliveros Colón, Luke Metz, Lütfi Kerem Şenel, Maarten Bosma, Maarten Sap, Maartje ter Hoeve, Maheen Farooqi, Manaal Faruqui, Mantas Mazeika, Marco Baturan, Marco Marelli, Marco Maru, Maria Jose Ramírez Quintana, Marie Tolkiehn, Mario Giulianelli, Martha Lewis, Martin Potthast, Matthew L. Leavitt, Matthias Hagen, Mátyás Schubert, Medina Orduna Baitemirova, Melody Arnaud, Melvin McElrath, Michael A. Yee, Michael Cohen, Michael Gu, Michael Ivanitskiy, Michael Starritt, Michael Strube, Michał Swędrowski, Michele Bevilacqua, Michihiro Yasunaga, Mihir Kale, Mike Cain, Mimee Xu, Mirac Suzgun, Mitch Walker, Mo Tiwari, Mohit Bansal, Moin Aminnaseri, Mor Geva, Mozhdeh Gheini, Mukund Varma T, Nanyun Peng, Nathan A. Chi, Nayeon Lee, Neta Gur-Ari Krakover, Nicholas Cameron, Nicholas Roberts, Nick Doiron, Nicole Martinez, Nikita Nangia, Niklas Deckers, Niklas Muennighoff, Nitish Shirish Keskar, Niveditha S. Iyer, Noah Constant, Noah Fiedel, Nuan Wen, Oliver Zhang, Omar Agha, Omar Elbaghdadi, Omer Levy, Owain Evans, Pablo Antonio Moreno Casares, Parth Doshi, Pascale Fung, Paul Pu Liang, Paul Vicol, Pegah Alipoormolabashi, Peiyuan Liao, Percy Liang, Peter Chang, Peter Eckersley, Phu Mon Htut, Pinyu Hwang, Piotr Miłkowski, Piyush Patil, Pouya Pezeshkpour, Priti Oli, Qiaozhu Mei, Qing Lyu, Qinlang Chen, Rabin Banjade, Rachel Etta Rudolph, Raefer Gabriel, Rahel Habacker, Ramon Risco, Raphaël Millière, Rhythm Garg, Richard Barnes, Rif A. Saurous, Riku Arakawa, Robbe Raymaekers, Robert Frank, Rohan Sikand, Roman Novak, Roman Sitelew, Ronan LeBras, Rosanne Liu, Rowan Jacobs, Rui Zhang, Ruslan Salakhutdinov, Ryan Chi, Ryan Lee, Ryan Stovall, Ryan Teehan, Rylan Yang, Sahib Singh, Saif M. Mohammad, Sajant Anand, Sam Dillavou, Sam Shleifer, Sam Wiseman, Samuel Gruetter, Samuel R. Bowman, Samuel S. Schoenholz, Sanghyun Han, Sanjeev Kwatra, Sarah A. Rous, Sarik Ghazarian, Sayan Ghosh, Sean Casey, Sebastian Bischoff, Sebastian Gehrmann, Sebastian Schuster, Sepideh Sadeghi, Shadi Hamdan, Sharon Zhou, Shashank Srivastava, Sherry Shi, Shikhar Singh, Shima Asaadi, Shixiang Shane Gu, Shubh Pachchigar, Shubham Toshniwal, Shyam Upadhyay, Shyamolima (Shammie) Debnath, Siamak Shakeri, Simon Thormeyer, Simone Melzi, Siva Reddy, Sneha Priscilla Makini, Soo-Hwan Lee, Spencer Torene, Sriharsha Hatwar, Stanislas Dehaene, Stefan Divic, Stefano Ermon, Stella Biderman, Stephanie Lin, Stephen Prasad, Steven T. Piantadosi, Stuart M. Shieber, Summer Misherghi, Svetlana Kiritchenko, Swaroop Mishra, Tal Linzen, Tal Schuster, Tao Li, Tao Yu, Tariq Ali, Tatsu Hashimoto, Te-Lin Wu, Théo Desbordes, Theodore Rothschild, Thomas Phan, Tianle Wang, Tiberius Nkinyili, Timo Schick, Timofei Kornev, Titus Tunduny, Tobias Gerstenberg, Trenton Chang, Trishala Neeraj, Tushar Khot, Tyler Shultz, Uri Shaham, Vedant Misra, Vera Demberg, Victoria Nyamai, Vikas Raunak, Vinay Ramasesh, Vinay Uday Prabhu, Vishakh Padmakumar, Vivek Srikumar, William Fedus, William Saunders, William Zhang, Wout Vossen, Xiang Ren, Xiaoyu Tong, Xinran Zhao, Xinyi Wu, Xudong Shen, Yadollah Yaghoobzadeh, Yair Lakretz, Yangqiu Song, Yasaman Bahri, Yejin Choi, Yichi Yang, Yiding Hao, Yifu Chen, Yonatan Belinkov, Yu Hou, Yufang Hou, Yuntao Bai, Zachary Seid, Zhuoye Zhao, Zijian Wang, Zijie J. Wang, Zirui Wang, Ziyi Wu

Abstract

Instruction finetuning and 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	BIG-bench + MMLU 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

Instruction finetuning and performance gains

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)

Instruction finetuning and performance gains

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.

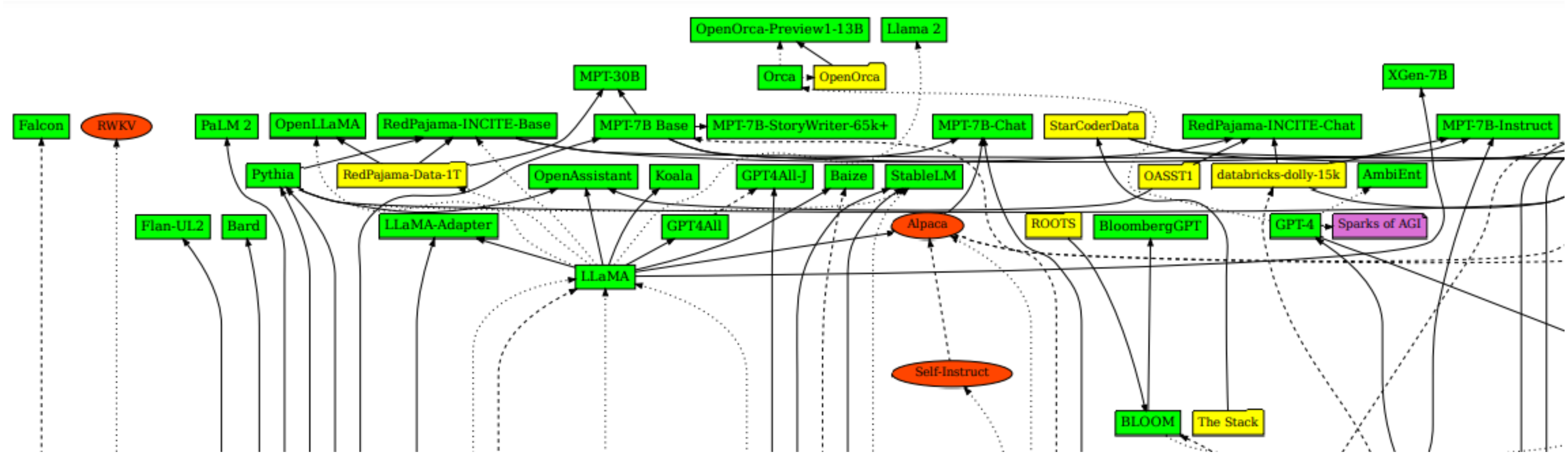
Options:
(A) They will discuss the reporter's favorite dishes
(B) They will discuss the chef's favorite dishes
(C) Ambiguous

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). ✓

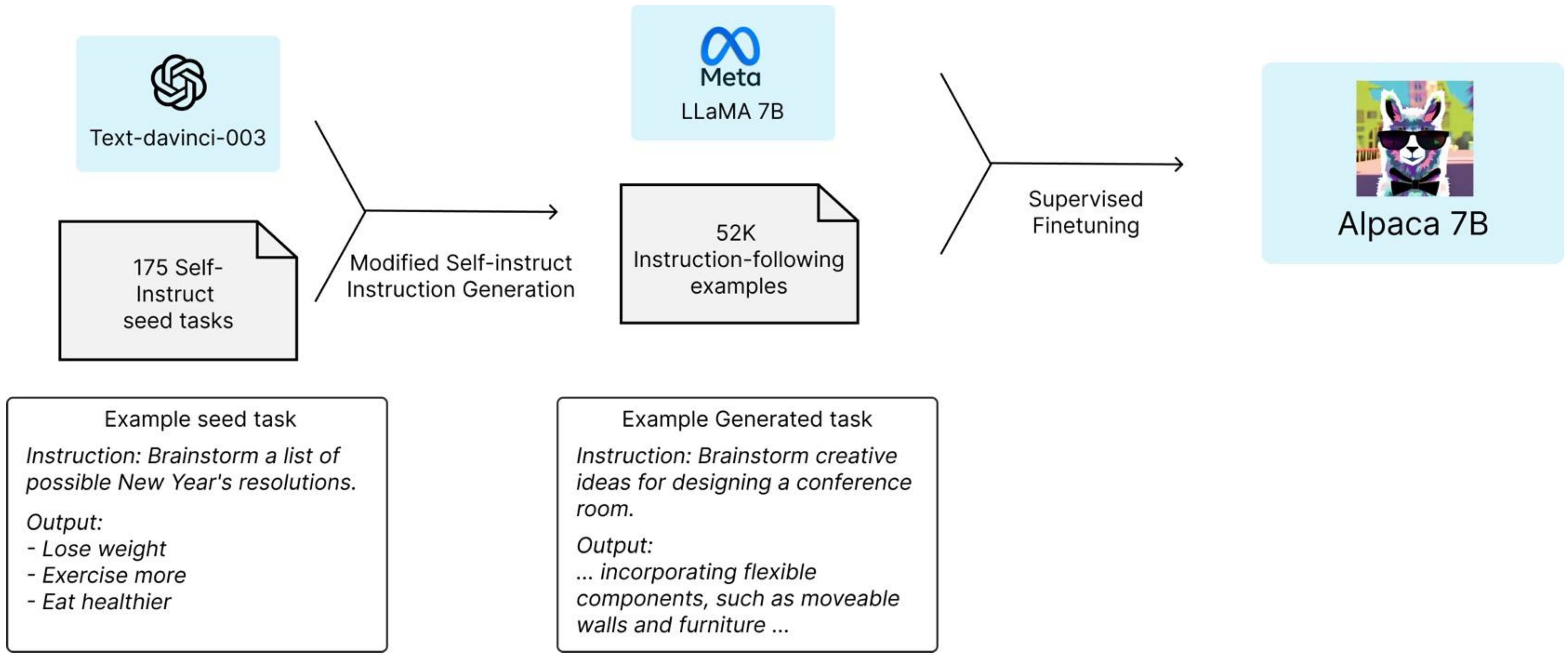
Cambrian explosion of instruction-tuning data



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

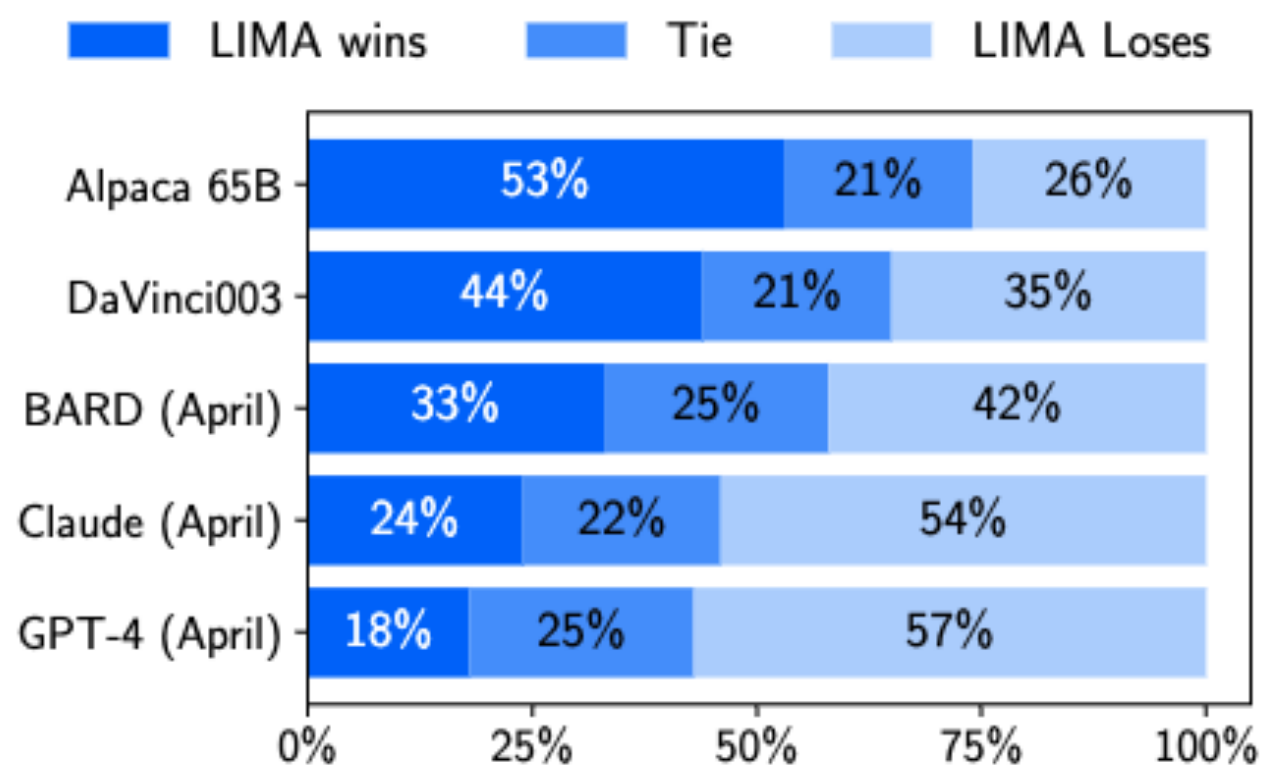
What have we learned?

- 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 Is 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



Reinforcement Learning from Human Feedback

Limitations of instruction tuning

Limitations of instruction tuning

- One limitation of instruction tuning is obvious: it's **expensive** to collect (and train on) ground-truth for lots of tasks

Limitations of instruction tuning

- One limitation of instruction tuning is obvious: it's **expensive** to collect (and train on) ground-truth for lots of tasks
- But there are other, subtler limitations, as well. Ideas?

Limitations of instruction tuning

- One limitation of instruction tuning is obvious: it's **expensive** to collect (and train on) ground-truth for lots of tasks
- But there are other, subtler limitations, as well. Ideas?
- Problem 1: open-ended generation tasks have no one right answer
  - *Write me a story about a fairy and a language model.*

Limitations of instruction tuning

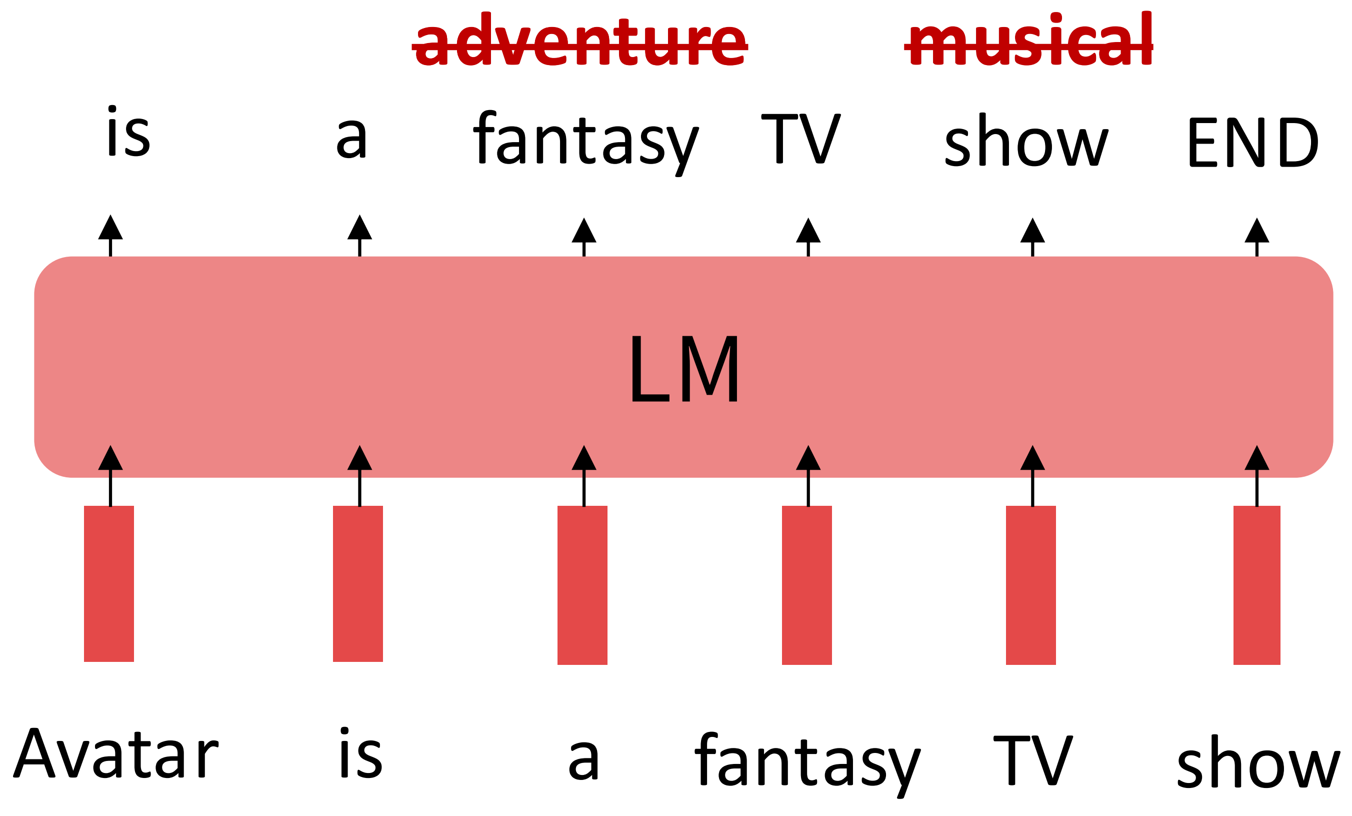
- One limitation of instruction tuning is obvious: it's **expensive** to collect (and train on) ground-truth for lots of tasks

- But there are other, subtler limitations, as well. Ideas?

- Problem 1: open-ended generation tasks have no one right answer

  - *Write me a story about a fairy and a language model.*

- Problem 2: maximum likelihood penalizes all token-level mistakes equally, but some errors are worse than others



Limitations of instruction tuning

- One limitation of instruction tuning is obvious: it's **expensive** to collect (and train on) ground-truth for lots of tasks

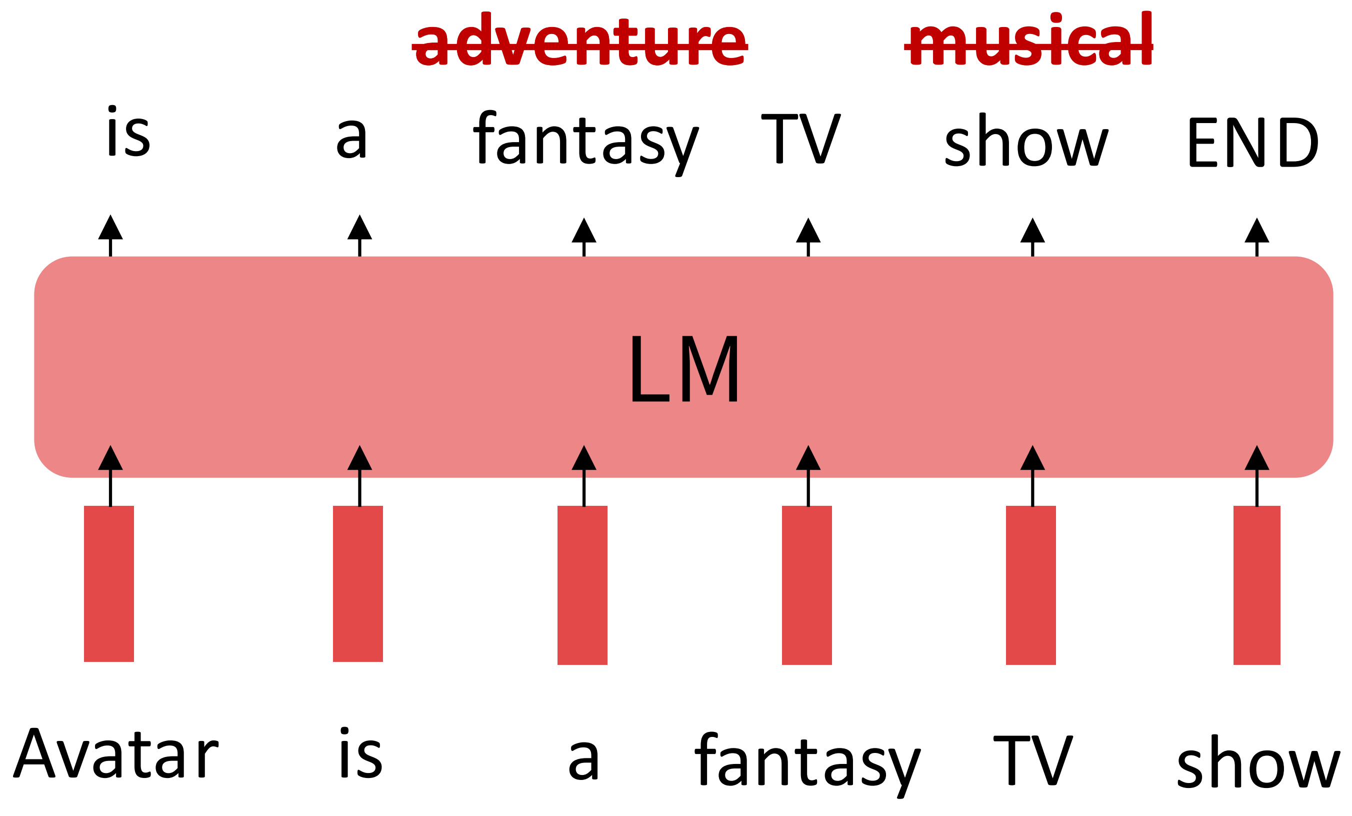
- But there are other, subtler limitations, as well. Ideas?

- Problem 1: open-ended generation tasks have no one right answer

  - *Write me a story about a fairy and a language model.*

- Problem 2: maximum likelihood penalizes all token-level mistakes equally, but some errors are worse than others

- So even instruction tuning isn't quite **matching human preferences**.



Limitations of instruction tuning

- One limitation of instruction tuning is obvious: it's **expensive** to collect (and train on) ground-truth for lots of tasks

- But there are other, subtler limitations, as well. Ideas?

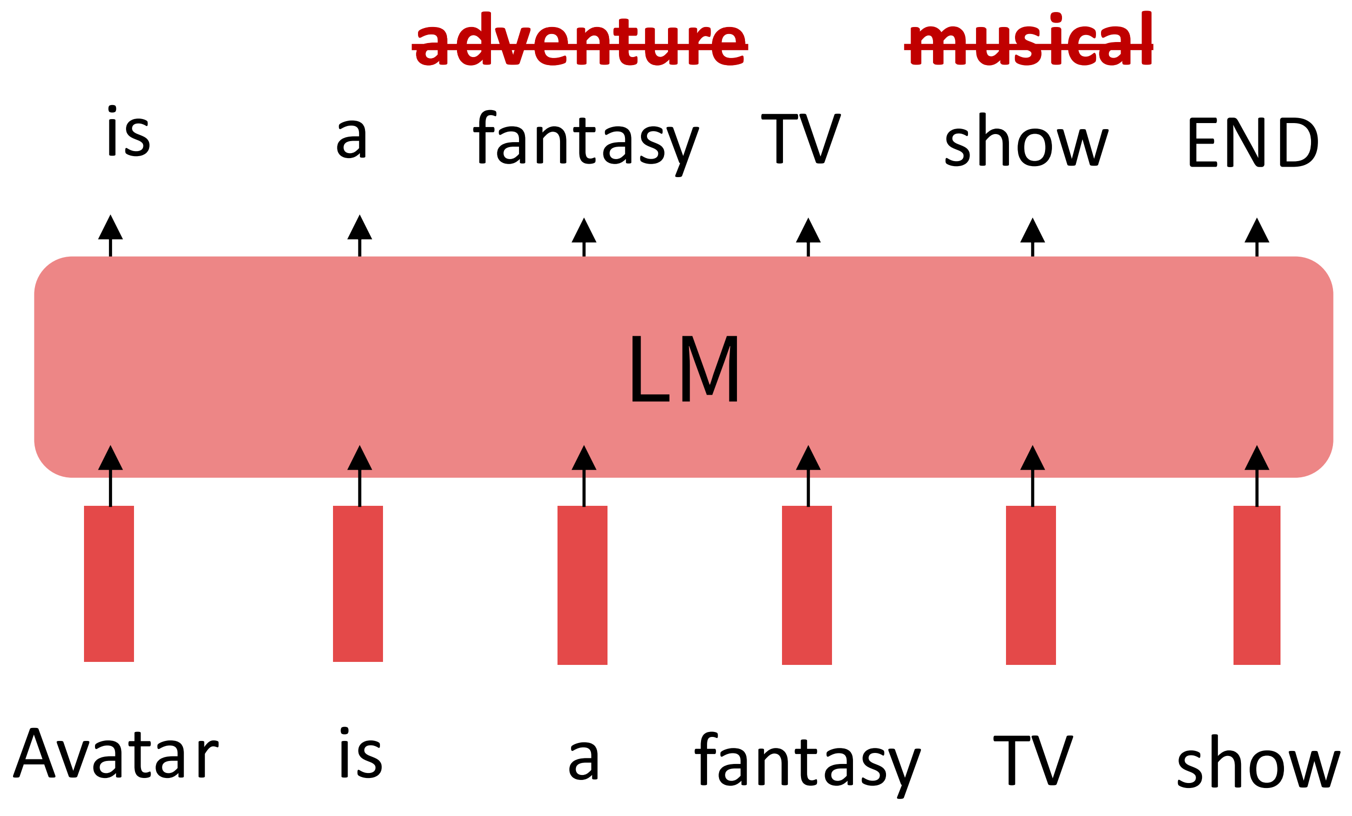
- Problem 1: open-ended generation tasks have no one right answer

  - *Write me a story about a fairy and a language model.*

- Problem 2: maximum likelihood penalizes all token-level mistakes equally, but some errors are worse than others

- So even instruction tuning isn't quite **matching human preferences**.

- So how can we do that?



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"

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"

Reinforcement Learning with Human Feedback!

Fine-Tuning Language Models from Human Preferences

Daniel M. Ziegler* **Nisan Stiennon*** **Jeffrey Wu** **Tom B. Brown**
Alec Radford **Dario Amodei** **Paul Christiano** **Geoffrey Irving**
OpenAI
{dmz,nisan,jeffwu,tom,alec,damodei,paul,irving}@openai.com

arxiv in Sep 2019
NeurIPS 2020

Learning to summarize from human feedback

Nisan Stiennon* **Long Ouyang*** **Jeff Wu*** **Daniel M. Ziegler*** **Ryan Lowe***

Chelsea Voss* **Alec Radford** **Dario Amodei** **Paul Christiano***

OpenAI

arxiv in Sep 2020
NeurIPS 2020

"Learning to Summarize with Human Feedback"

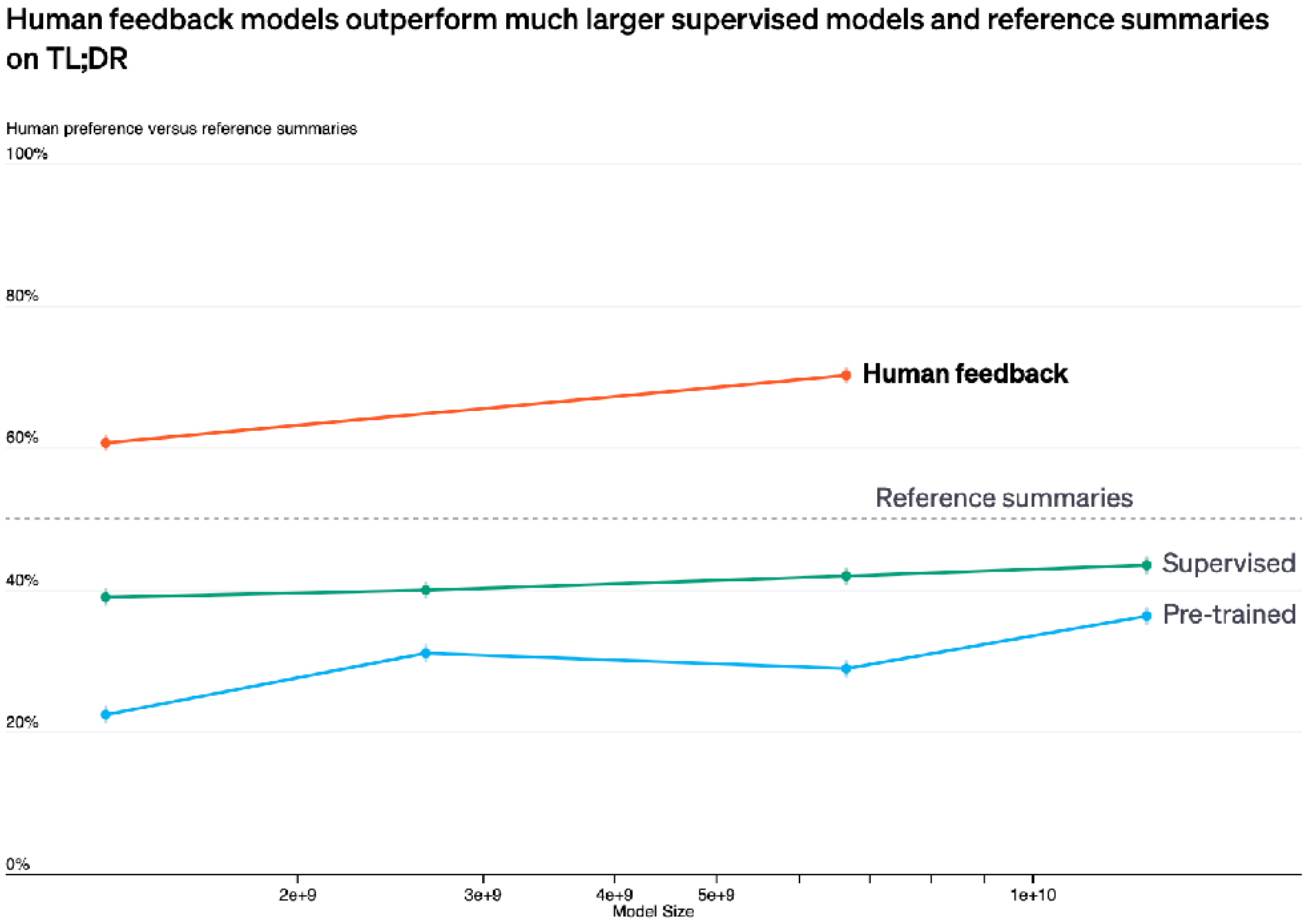
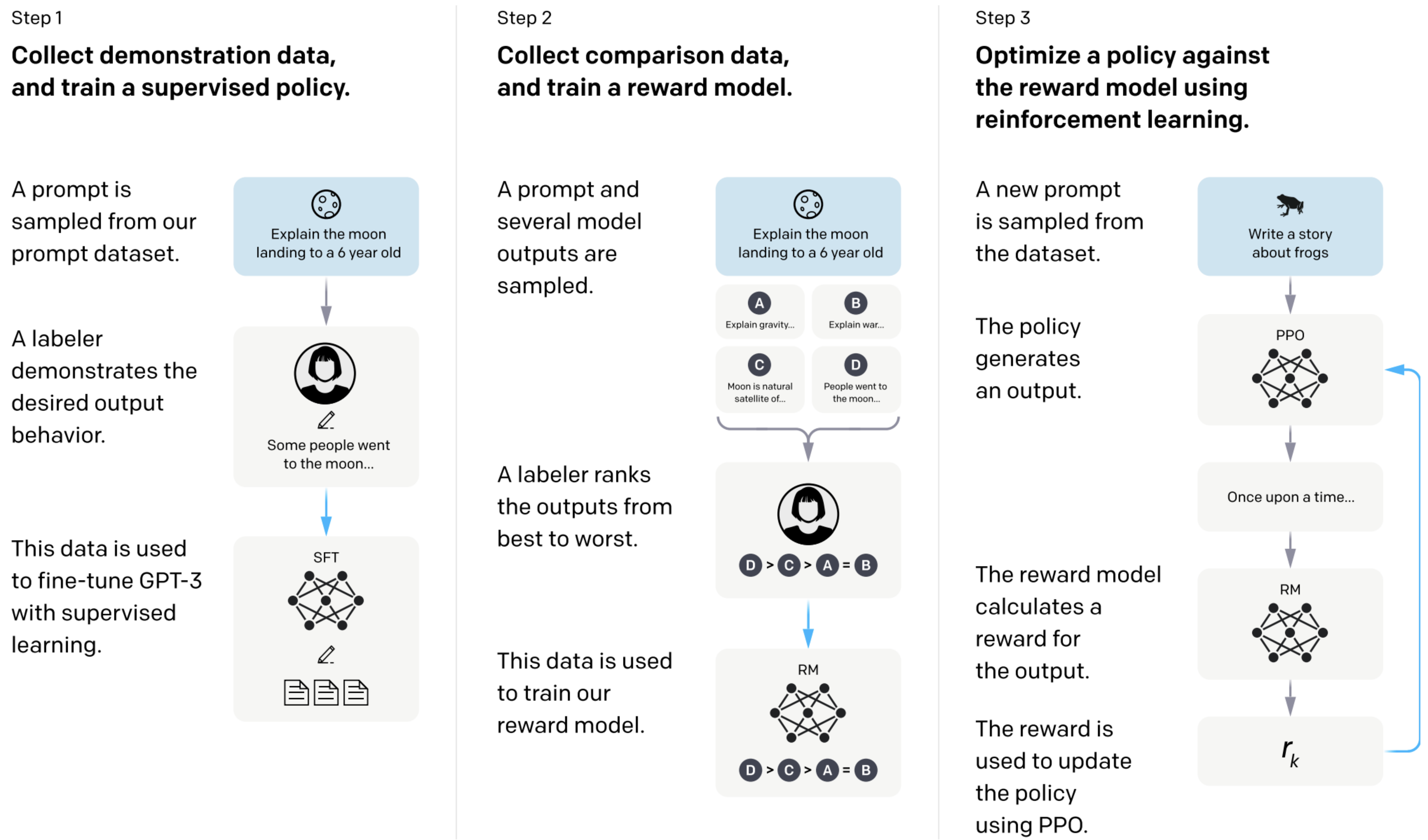


Figure 1: The performance of various training procedures for different model sizes. Model performance is measured by how often summaries from that model are preferred to the human-written reference summaries. Our pre-trained models are early versions of GPT-3, our supervised baselines were fine-tuned to predict 117K human-written TL;DRs, and our human feedback models are additionally fine-tuned on a dataset of about 65K summary comparisons.

https://openai.com/research/learning-to-summarize-with-human-feedback

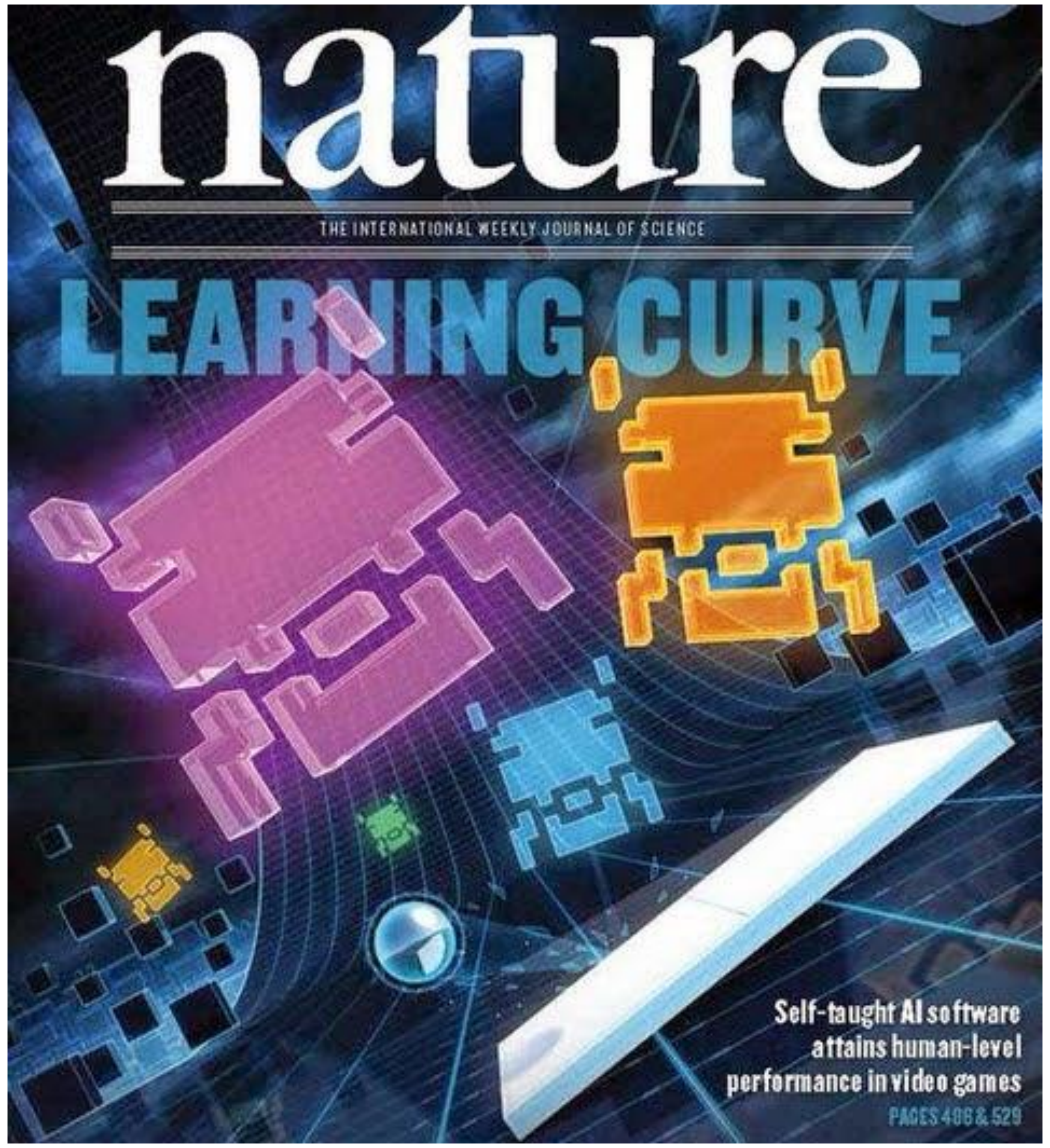
Overview of RLHF



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

Resurgence in Reinforcement Learning

- The field of **reinforcement learning (RL)** has studied these (and related) problems for many years now [Williams, 1992; Sutton and Barto, 1998]
- Circa 2013: resurgence of interest in RL applied to deep learning, game-playing [Mnih et al., 2013]
- 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

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!

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

How do we estimate this expectation??

What if our reward function is non-differentiable??

- **Policy gradient** methods in RL (e.g., REINFORCE; [Williams, 1992]) give us tools for estimating and optimizing this objective.

Early work at Northeastern! Ronald Williams *retired* before I got here over a decade ago, so RL has been developing for a while.

A Sketch of REINFORCE (Williams, 1992)

- We want to obtain

(defn. of expectation) (linearity of gradient)

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

- Here we'll use a very handy trick known as the **log-derivative trick**. Let's try taking the gradient of $\log p_\theta(s)$

$$\nabla_\theta \log p_\theta(s) = \frac{1}{p_\theta(s)} \nabla_\theta p_\theta(s) \quad \Longrightarrow \quad \nabla_\theta p_\theta(s) = p_\theta(s)\ \nabla_\theta \log p_\theta(s)$$

(chain rule)

- Plug back in:

This is an expectation of this

$$\sum_s R(s)\, \nabla_\theta p_\theta(s) = \sum_s p_\theta(s) R(s)\ \nabla_\theta \log p_\theta(s)$$

A Sketch of REINFORCE (Williams, 1992)

- Now we have put the gradient "inside" the expectation, we can approximate this objective with Monte Carlo samples:

$$\nabla_\theta \mathbb{E}_{\hat{s}\sim p_\theta(s)}[R(\hat{s})] = \mathbb{E}_{\hat{s}\sim p_\theta(s)}[R(\hat{s})\, \nabla_\theta \log p_\theta(\hat{s})] \approx \frac{1}{m}\sum_{i=1}^{m} R(s_i)\, \nabla_\theta \log p_\theta(s_i)$$

This is why it's called **"reinforcement learning"**: we **reinforce** good actions, increasing the chance they happen again.

If R is +++

Take gradient steps to maximize $p_\theta(s_i)$

- Giving us the update rule: $\theta_{t+1} \coloneqq \theta_t + \alpha \frac{1}{m}\sum_{i=1}^{m} R(s_i)\, \nabla_{\theta_t} \log p_{\theta_t}(s_i)$

If R is ---

Take steps to minimize $p_\theta(s_i)$

This is **heavily simplified**! There is a *lot* more needed to do RL w/ LMs. **Can you see any problems with this objective?**

What's the reward?

- 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]

```
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

$R(s_2) = 1.2$

Train an LM $RM_\phi(s)$ to predict human preferences from an annotated dataset, then optimize for RM_ϕ instead.

What's the reward?

- **Problem 2:** human judgments are noisy and miscalibrated!
- **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.
```

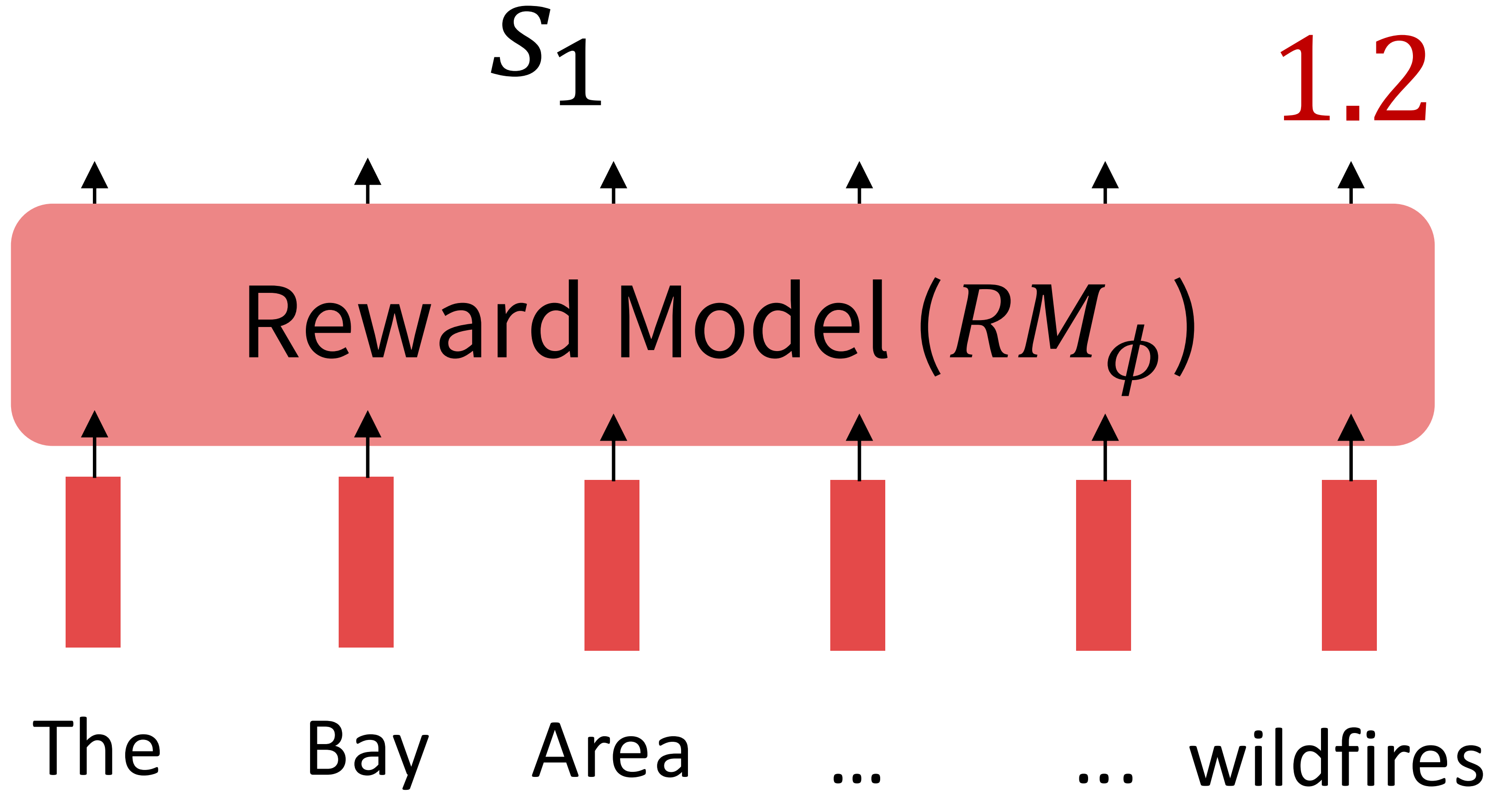
s_3

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

What's the reward?

- **Problem 2:** human judgments are noisy and miscalibrated!
- **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_1		s_3		s_2



Bradley-Terry [1952] paired comparison model

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

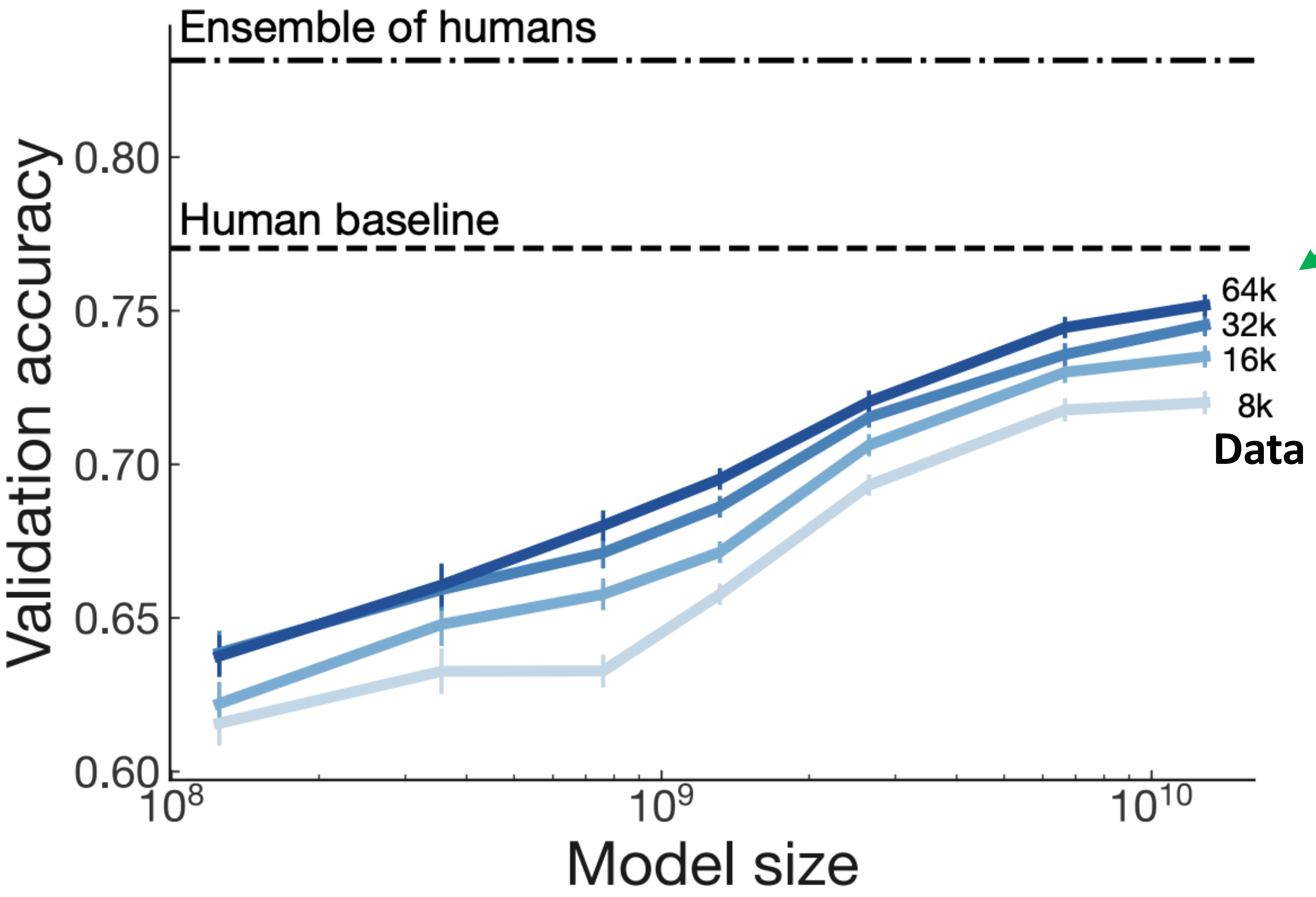
"winning" sample

"losing" sample

s^w should score higher than s^l

Evaluate the reward model

Evaluate RM on predicting outcome of held-out human judgments



Large enough RM trained on enough data approaching single human perf

[Stiennon et al., 2020]

RLHF: Putting the pieces together

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

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

RLHF: Putting the pieces together

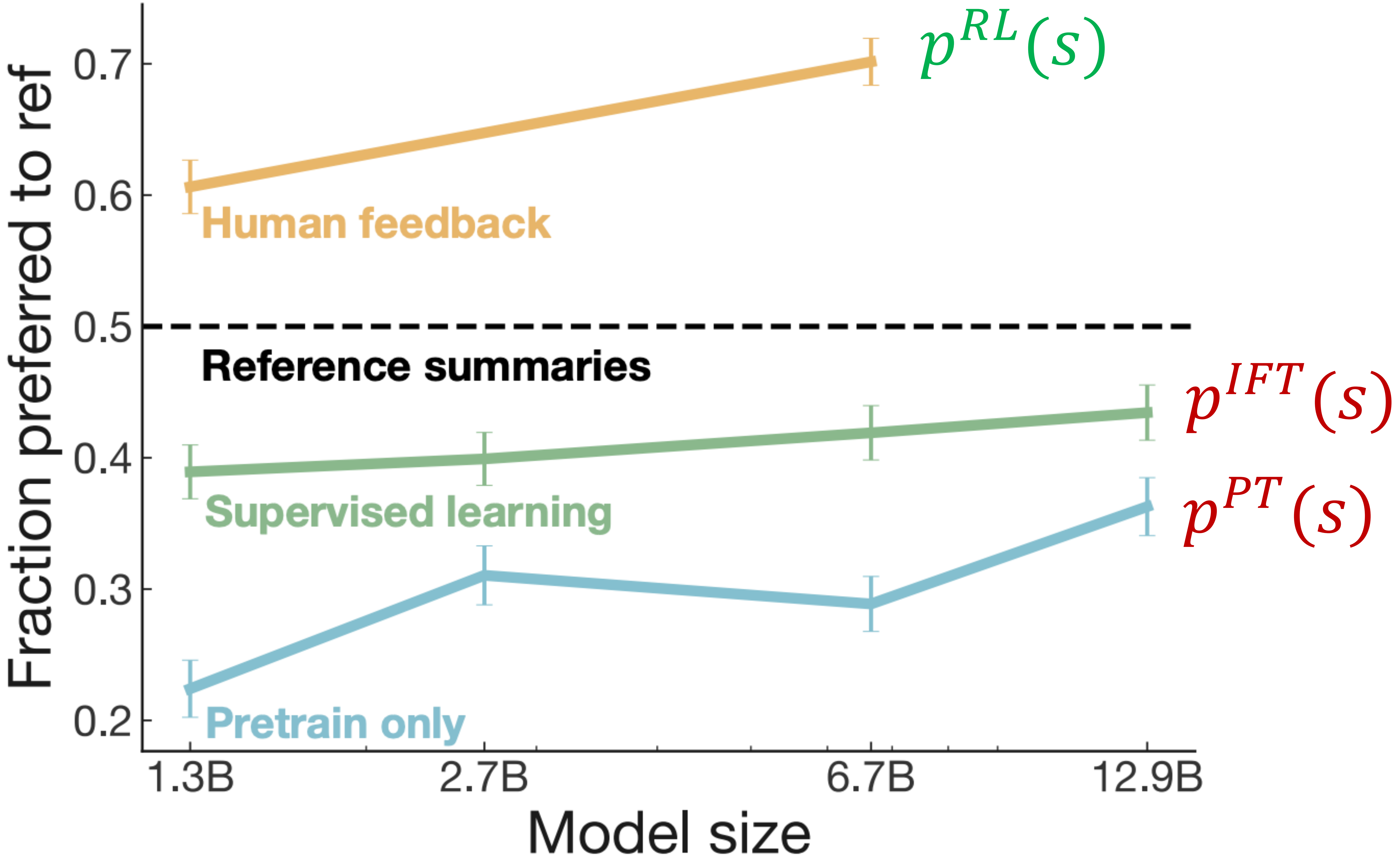
- Finally, we have everything we need:
  - A pretrained (possibly instruction-finetuned) LM $p^{PT}(s)$
  - A reward model $RM_\phi(s)$ that produces scalar rewards for LM outputs, trained on a dataset of human comparisons
  - A method for optimizing LM parameters towards an arbitrary reward function.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

$$R(s) = RM_\phi(s) - \underbrace{\beta \log\left(\frac{p_\theta^{RL}(s)}{p^{PT}(s)}\right)}_{\text{This is a penalty which prevents us from diverging too far from the pretrained model.}}$$

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

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

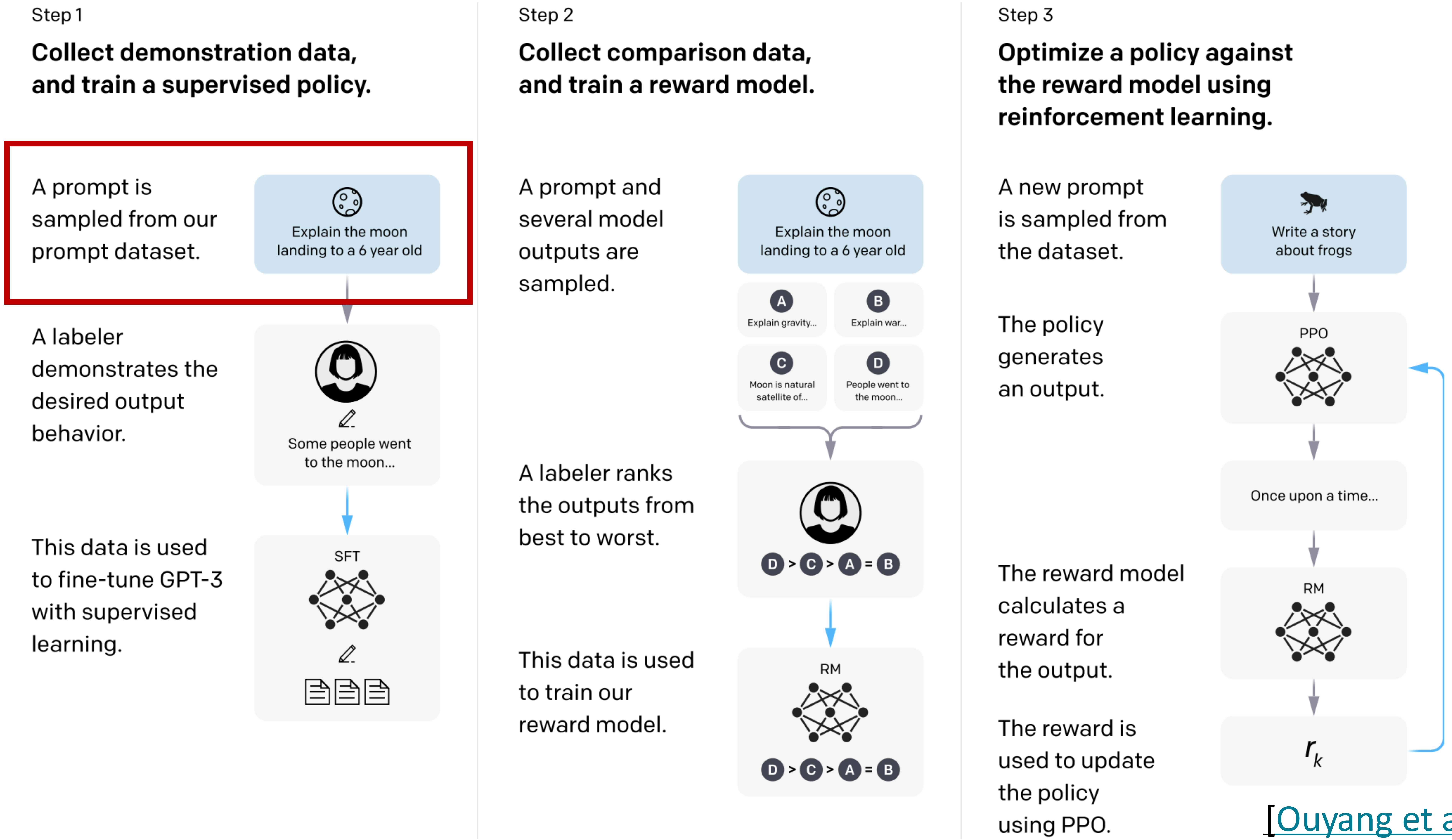
RLHF: Gains over retraining and fine-tuning



[Stiennon et al., 2020]

InstructGPT and ChatGPT

InstructGPT: Scaling to 30k tasks



[Ouyang et al., 2022]

InstructGPT: Scaling to 30k tasks

Tasks collected from labelers:

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

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

[Ouyang et al., 2022]

InstructGPT

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

COMPLETION

GPT-3

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

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

Explain the big bang theory to a 6 year old.

Explain evolution to a 6 year old.
```

InstructGPT

```
People went to the moon, and they took pictures of what they saw, and
sent them back to the earth so we could all see them.
```

ChatGPT: Instruction tuning + RLHF for dialog

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!)

https://openai.com/blog/chatgpt/

ChatGPT: Instruction tuning + RLHF for dialog

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!)

https://openai.com/blog/chatgpt/

What does RLHF change?

> **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

Beyond RLHF

Limitations of RL + reward models

- Human preferences are unreliable!
  - ”Reward hacking” is a common problem in RL
  - Chatbots are rewarded to produce responses that *seem* authoritative and helpful, *regardless of truth*
  - This can result in making up facts + hallucinations

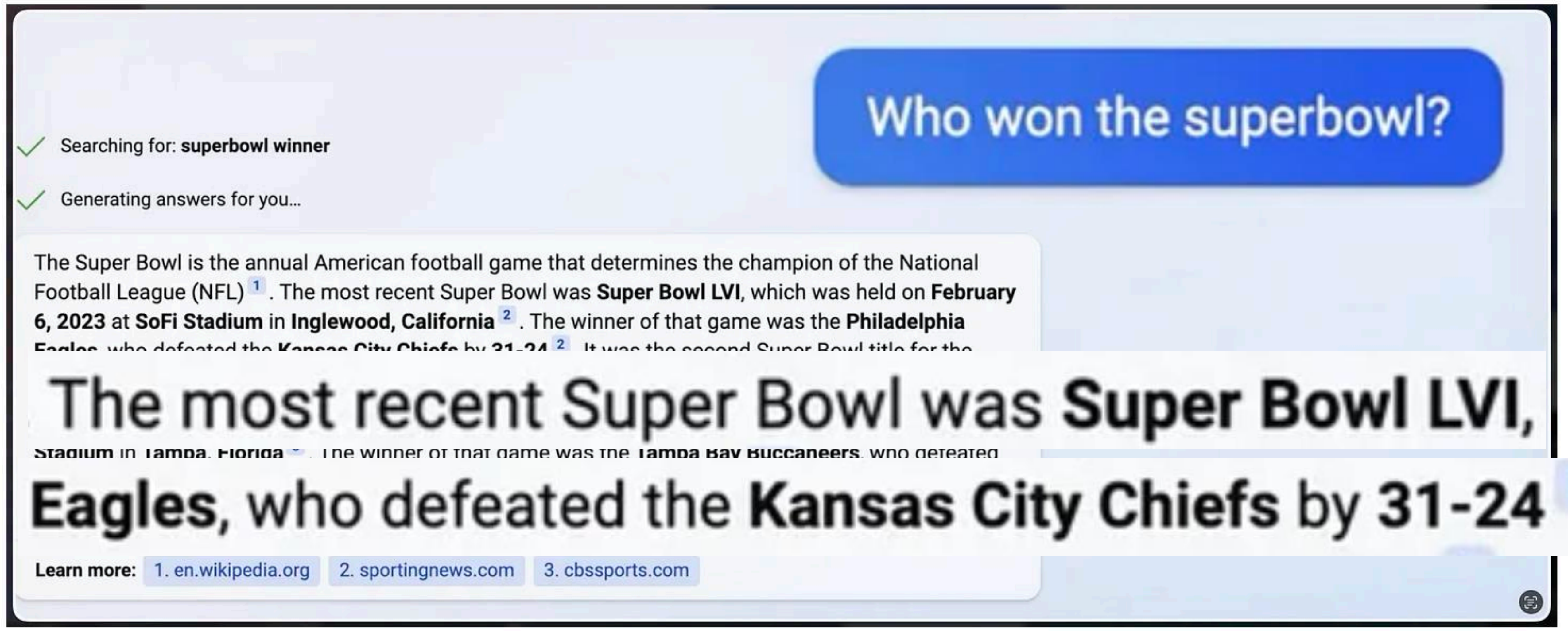
TECHNOLOGY

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

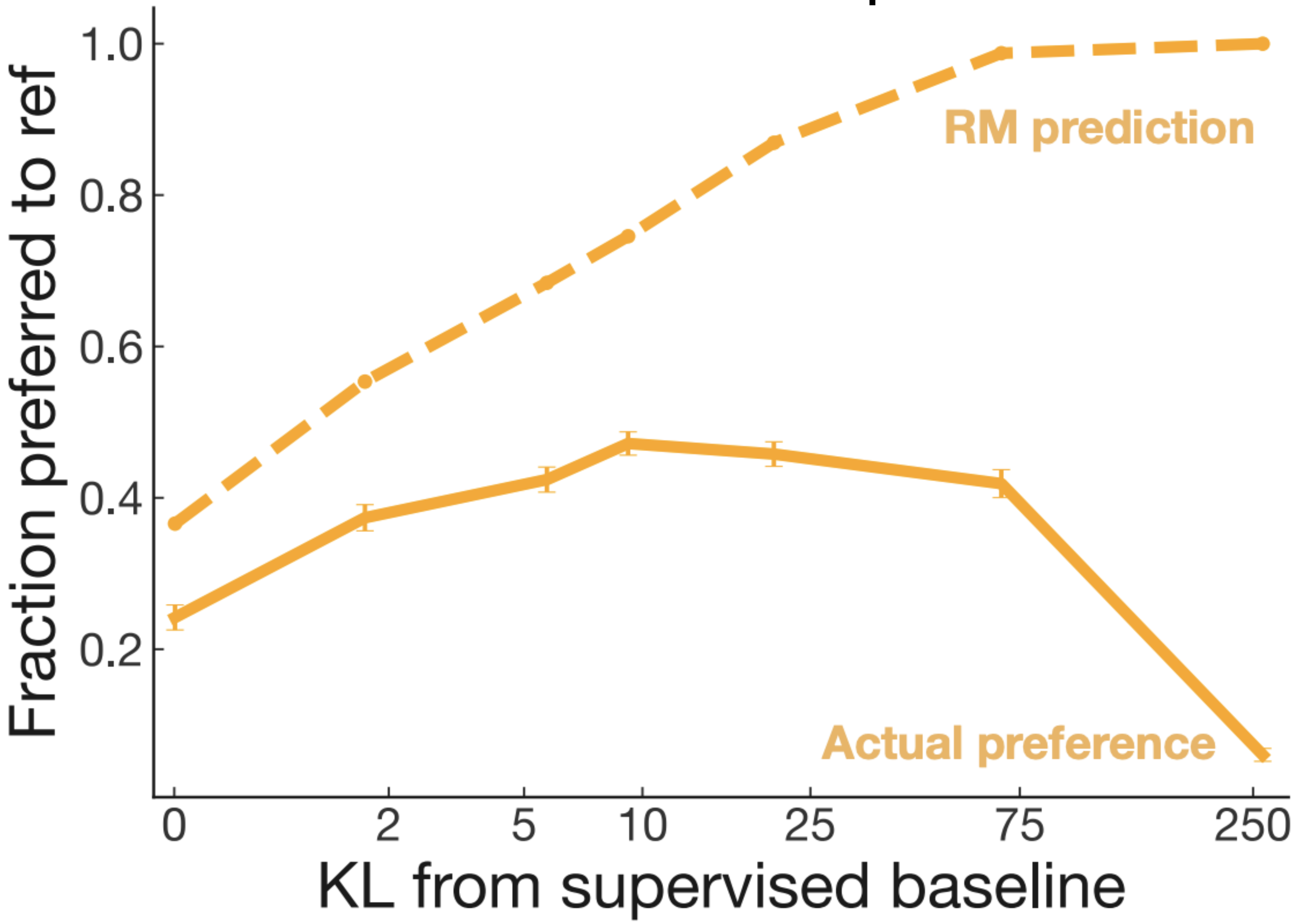


https://news.ycombinator.com/item?id=34776508
https://apnews.com/article/kansas-city-chiefs-philadelphia-eagles-technology-science-82bc20f207e3e4cf81abc6a5d9e6b23a

Limitations of RL + reward models

- Human preferences are unreliable!
  - ”Reward hacking” is a common problem in RL
  - 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 model over-optimization



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

[Stiennon et al., 2020]

Direct Preference Optimization

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:

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

Peters & Schaal 2007

- 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 any arbitrary LMs, thus

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

DPO: Putting the pieces together

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

- Final DPO loss via the Bradley-Terry model of human preferences:

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

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

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

Reward for winning sample

Reward for losing sample

[Rafailov+ 2023]

DPO: Putting the pieces together

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

- Final DPO loss via the Bradley-Terry model of human preferences:

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

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

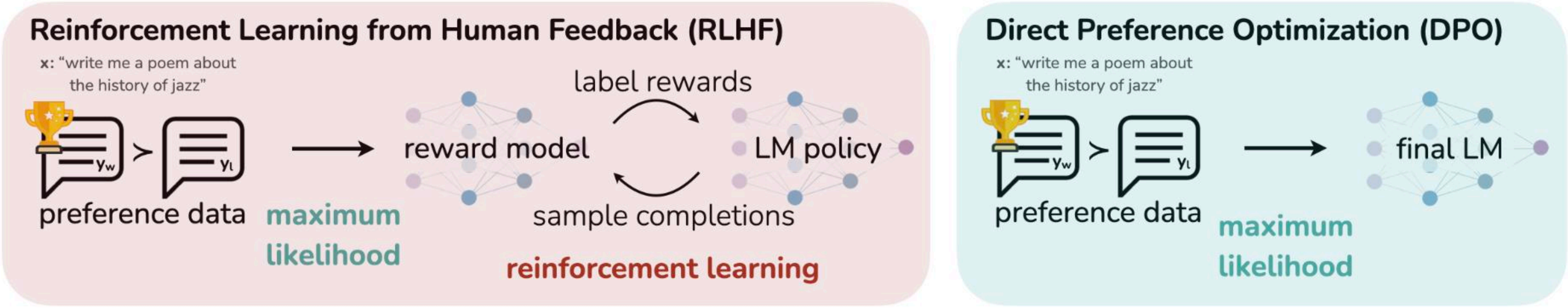
$$= -\mathbb{E}_{(x,y_w,y_l)\sim D}\left[\log \sigma(\beta \log \frac{p_\theta^{RL}(y_w|x)}{p^{PT}(y_w|x)} - \beta \log \frac{p_\theta^{RL}(y_l|x)}{p^{PT}(y_l|x)})\right]$$

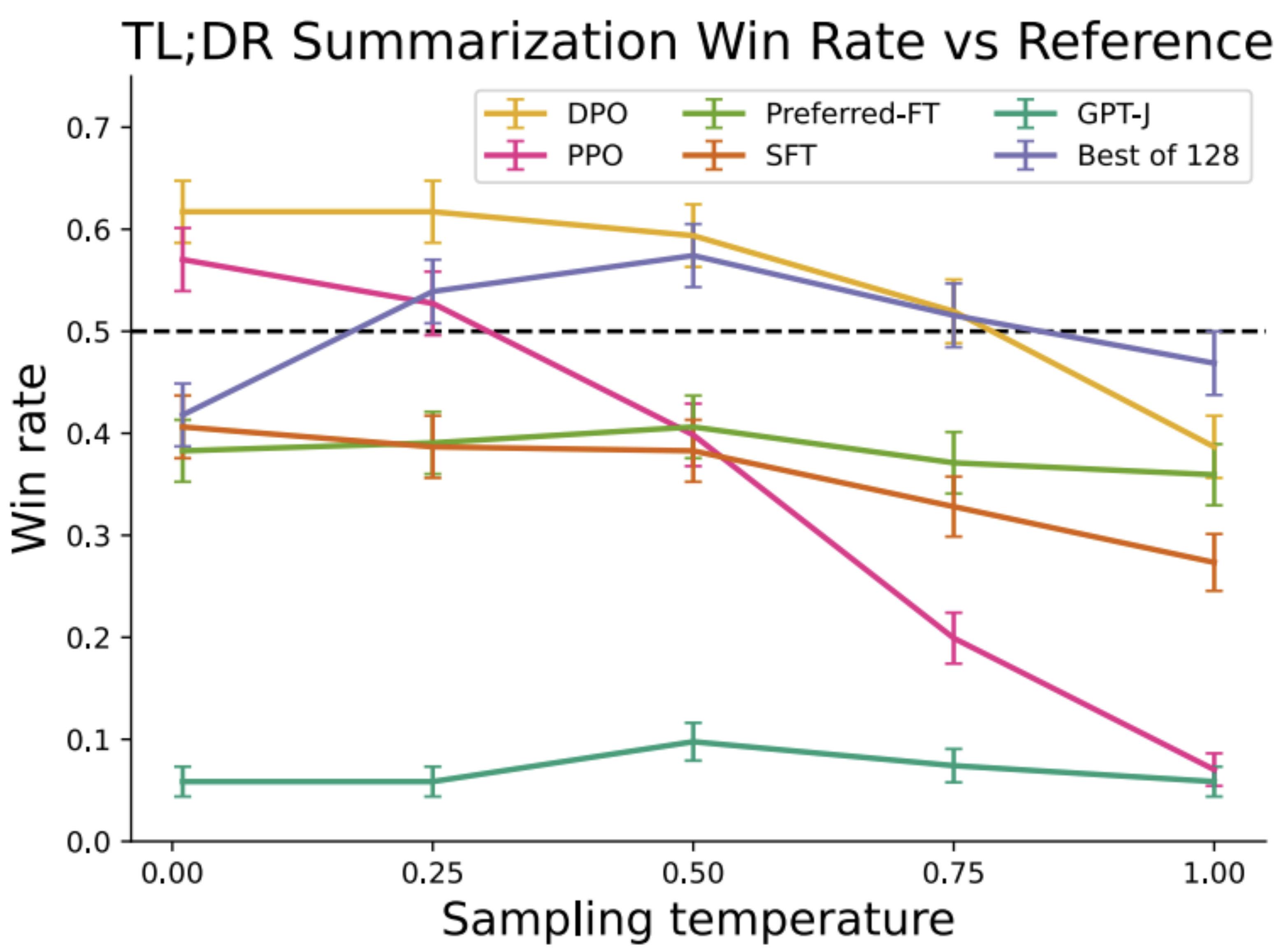
Reward for winning sample

Reward for losing sample

[Rafailov+ 2023]

DPO outperforms prior methods





- You can replace the complex RL part with a very simple weighted MLE objective
- Other variants (KTO, IPO) now emerging too
- TL;DR summarization win rates vs. human-written summaries (GPT-4 as a judge)

Scaling DPO Performance

	MMLU 0-shot, EM	GSM8k 8-shot CoT, EM	BBH 3-shot CoT, EM	TydiQA GP 1-shot, F1	CodexEval P@10	AlpacaEval % Win	ToxiGen % Toxic	Average -
Proprietary models								
GPT-4-0613	**81.4**	**95.0**	**89.1**	**65.2**	87.0	91.2	0.6	**86.9**
GPT-3.5-turbo-0613	65.7	76.5	70.8	51.2	88.0	**91.8**	**0.5**	77.6
GPT-3.5-turbo-0301	67.9	76.0	66.1	51.9	**88.4**	83.6	27.7	72.3
Non-TÜLU Open Models								
Zephyr-Beta 7B	58.6	28.0	44.9	23.7	54.3	86.3	64.0	47.4
Xwin-LM v0.1 70B	**65.0**	**65.5**	**65.6**	38.2	**66.1**	**95.8**	12.7	**69.1**
LLAMA-2-Chat 7B	46.8	12.0	25.6	22.7	24.0	87.3	**0.0**	45.4
LLAMA-2-Chat 13B	53.2	9.0	40.3	32.1	33.1	91.4	**0.0**	51.3
LLAMA-2-Chat 70B	60.9	59.0	49.0	**44.4**	52.1	94.5	**0.0**	65.7
TÜLU 2 Suite								
TÜLU 2 7B	50.4	34.0	48.5	46.4	36.9	73.9	7.0	54.7
TÜLU 2+DPO 7B	50.7	34.5	45.5	44.5	40.0	85.1	0.5	56.3
TÜLU 2 13B	55.4	46.0	49.5	53.2	49.0	78.9	1.7	61.5
TÜLU 2+DPO 13B	55.3	49.5	49.4	39.7	48.9	89.5	1.1	61.6
TÜLU 2 70B	67.3	**73.0**	**68.4**	**53.6**	68.5	86.6	0.5	**73.8**
TÜLU 2+DPO 70B	**67.8**	71.5	66.0	35.8	**68.9**	**95.1**	**0.2**	72.1

- Tulu2 has shown that it is possible to DPO a 70B base model, with good results.
- No comparison with PPO yet.