Reinforcement Learning from Human Feedback (RLHF)

Yu Meng
University of Virginia
yumeng5@virginia.edu

Nov 06, 2024

Announcement

- First guest lecture this Friday (11/08)!
- We’ll meet on Zoom (https://virginia.zoom.us/j/8397490876); no need to come to the classroom!
- We’ll take attendance on Zoom
  - You’ll get 1% participation credit for attending the guest lecture
  - Make sure your full name on Zoom matches your name on Canvas!
- You are encouraged to ask questions related to the talk!
  - You’ll get another 1% participation credit if you ask a question (even if it does not get answered due to time constraints)
  - You can either ask directly during the talk or type your question in the Zoom chat (we count both), but we won’t be using Slido for guest lectures

Overview of Course Contents

- Week 1: Logistics & Overview
- Week 2: N-gram Language Models
- Week 3: Word Senses, Semantics & Classic Word Representations
- Week 4: Word Embeddings
- Week 5: Sequence Modeling and Neural Language Models
- Week 6-7: Language Modeling with Transformers (Pretraining + Fine-tuning)
- Week 8: Large Language Models (LLMs) & In-context Learning
- Week 9-10: Reasoning, Knowledge, and Retrieval-Augmented Generation (RAG)
- Week 11: LLM Alignment
- Week 12: Language Agents
- Week 13: Recap + Future of NLP
- Week 15 (after Thanksgiving): Project Presentations

(Recap) Overview: Language Model Alignment

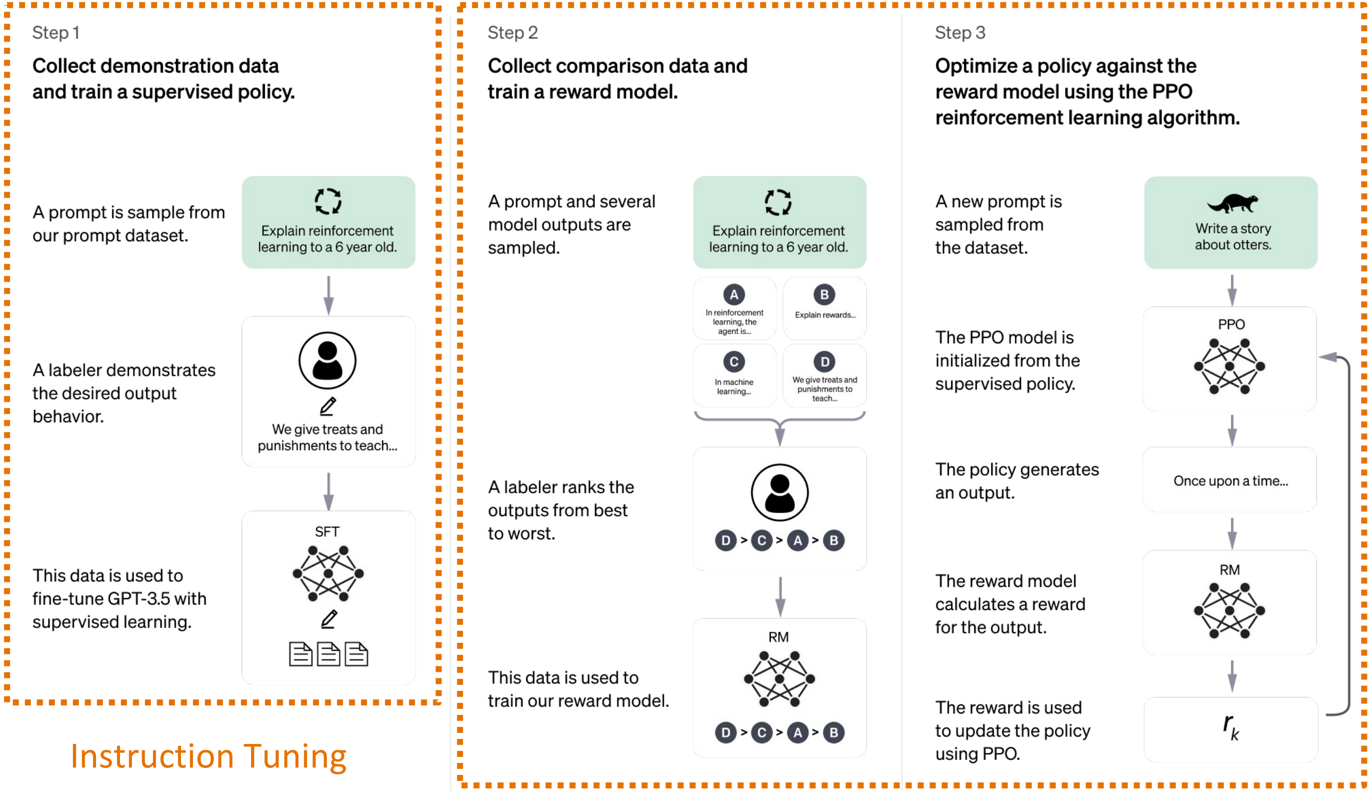
- Ensure language models behaviors are aligned with human values and intent
- “HHH” criteria (Askell et al. 2021):
  - **Helpful**: Efficiently perform the task requested by the user
  - **Honest**: Give accurate information & express uncertainty
  - **Harmless**: Avoid offensive/discriminatory/biased outputs

Paper: https://arxiv.org/pdf/2112.00861

(Recap) Post-training for Alignment

- Pretrained language models are **not** aligned
- Objective mismatch
  - Pretraining is to predict the next word in a sentence
  - Does not involve understanding human intent/values
- Training data bias
  - Text from the internet can contain biased, harmful, or misleading information
  - LMs don’t distinguish between good and bad behavior in training data
- (Over-)generalization issues
  - LMs’ generalization can lead to outputs that are inappropriate in specific contexts
  - Might not align with intended ethics/honesty standard

(Recap) Language Model Alignment Techniques



Step 1

Collect demonstration data and train a supervised policy.

A prompt is sample from our prompt dataset.

A labeler demonstrates the desired output behavior.

This data is used to fine-tune GPT-3.5 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 the PPO reinforcement learning algorithm.

A new prompt is sampled from the dataset.

The PPO model is initialized from the supervised policy.

The policy generates an output.

The reward model calculates a reward for the output.

The reward is used to update the policy using PPO.

Explain reinforcement learning to a 6 year old.
We give treats and punishments to teach...
SFT
Explain reinforcement learning to a 6 year old.
A
In reinforcement learning, the agent is...
B
Explain rewards...
C
In machine learning...
D
We give treats and punishments to teach...
D > C > A > B
RM
D > C > A > B
Write a story about otters.
PPO
Once upon a time...
RM
r_k

Instruction Tuning

Reinforcement Learning from Human Feedback (RLHF)

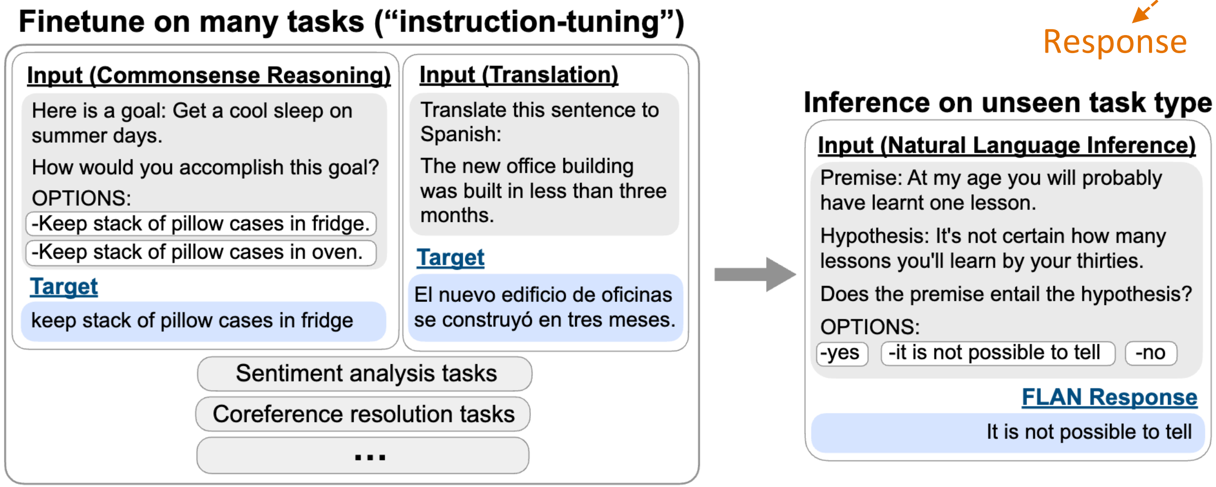
Figure source: https://openai.com/index/chatgpt/

(Recap) Instruction Tuning: Method

- **Input**: task description
- **Output**: expected response or solution to the task
- Train LLMs to generate response tokens given prompts $\min_{\boldsymbol{\theta}} -\log p_{\boldsymbol{\theta}}(\boldsymbol{y}|\boldsymbol{x})$

Response

Prompt



Finetune on many tasks ("instruction-tuning")

Input (Commonsense Reasoning)

Here is a goal: Get a cool sleep on summer days.

How would you accomplish this goal?

OPTIONS:

-Keep stack of pillow cases in fridge.

-Keep stack of pillow cases in oven.

Target

keep stack of pillow cases in fridge

Input (Translation)

Translate this sentence to Spanish:

The new office building was built in less than three months.

Target

El nuevo edificio de oficinas se construyó en tres meses.

Sentiment analysis tasks

Coreference resolution tasks

...

Inference on unseen task type

Input (Natural Language Inference)

Premise: At my age you will probably have learnt one lesson.

Hypothesis: It's not certain how many lessons you'll learn by your thirties.

Does the premise entail the hypothesis?

OPTIONS:

-yes -it is not possible to tell -no

FLAN Response

It is not possible to tell

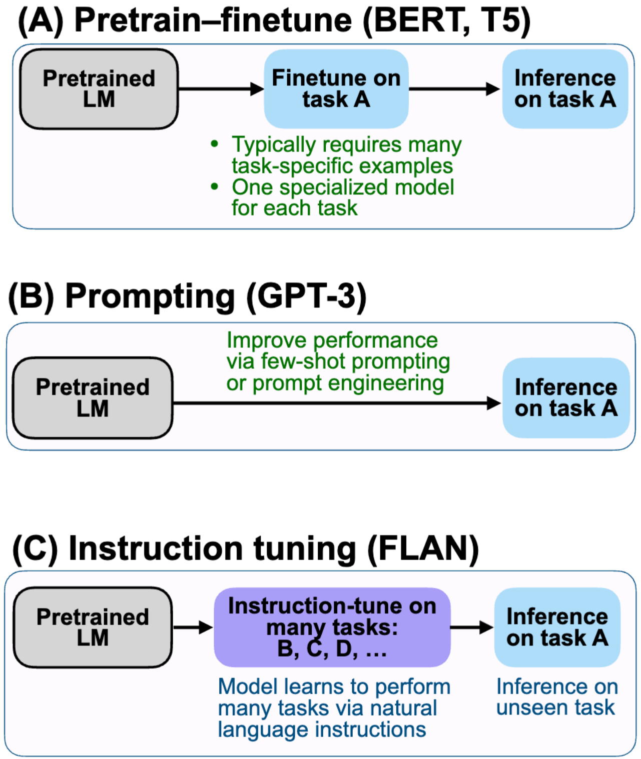
Paper: https://arxiv.org/pdf/2109.01652

(Recap) Instruction Tuning vs. Other Paradigms

- Task-specific fine-tuning does not enable generalization across multiple tasks

- In-context learning requires few-shot demonstrations

- Instruction tuning enables zero-shot cross task generalization

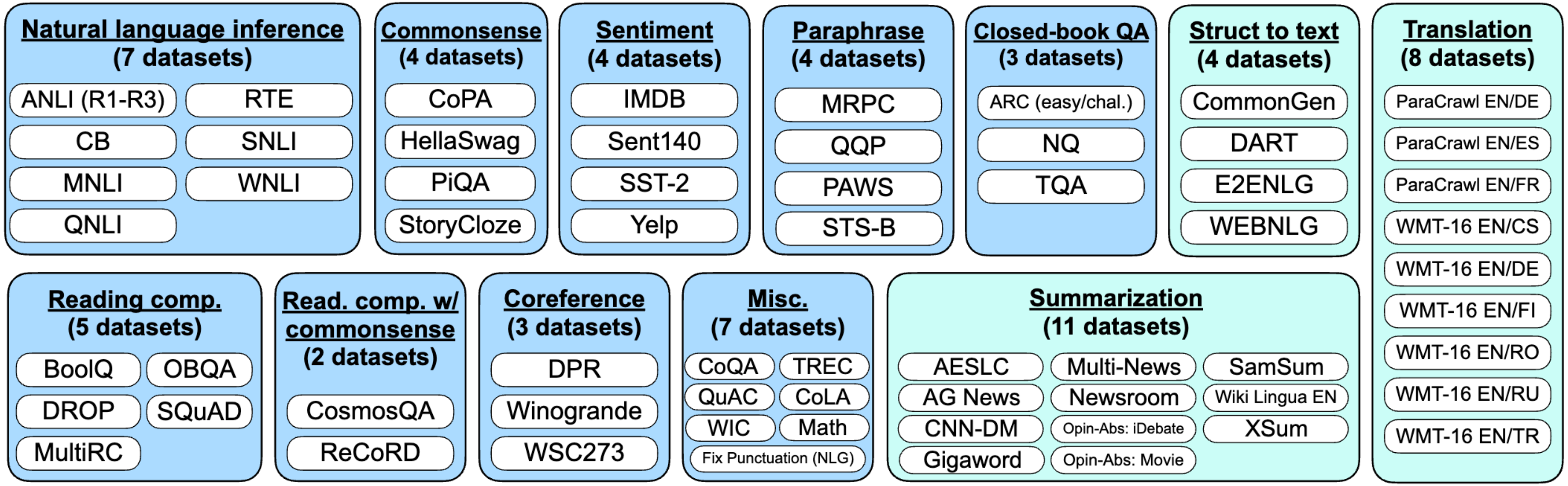

(A) Pretrain–finetune (BERT, T5)
Pretrained LM
Finetune on task A
Inference on task A
• Typically requires many task-specific examples
• One specialized model for each task
(B) Prompting (GPT-3)
Pretrained LM
Improve performance via few-shot prompting or prompt engineering
Inference on task A
(C) Instruction tuning (FLAN)
Pretrained LM
Instruction-tune on many tasks: B, C, D, …
Inference on task A
Model learns to perform many tasks via natural language instructions
Inference on unseen task

(Recap) Instruction Tuning vs. Pretraining

- Both instruction tuning and pretraining are **multi-task** learning paradigms
- Supervision
  - Pretraining: self-supervised learning (raw data w/o human annotation)
  - Instruction tuning: supervised learning (human annotated responses)
- Task format
  - Pretraining: tasks are implicit (predicting next tokens)
  - Instruction tuning: tasks are explicit (defined using natural language instructions)
- Goal
  - Pretraining: teach LMs a wide range of linguistic patterns & general knowledge
  - Instruction tuning: teach LMs to follow specific instructions and perform a variety of tasks

FLAN: Collection of Instruction Tuning Datasets

62 datasets (12 task clusters) covering a wide range of understanding + generation tasks



Natural language inference (7 datasets)
ANLI (R1-R3), RTE, CB, SNLI, MNLI, WNLI, QNLI

Commonsense (4 datasets)
CoPA, HellaSwag, PiQA, StoryCloze

Sentiment (4 datasets)
IMDB, Sent140, SST-2, Yelp

Paraphrase (4 datasets)
MRPC, QQP, PAWS, STS-B

Closed-book QA (3 datasets)
ARC (easy/chal.), NQ, TQA

Struct to text (4 datasets)
CommonGen, DART, E2ENLG, WEBNLG

Translation (8 datasets)
ParaCrawl EN/DE, ParaCrawl EN/ES, ParaCrawl EN/FR, WMT-16 EN/CS, WMT-16 EN/DE, WMT-16 EN/FI, WMT-16 EN/RO, WMT-16 EN/RU, WMT-16 EN/TR

Reading comp. (5 datasets)
BoolQ, OBQA, DROP, SQuAD, MultiRC

Read. comp. w/ commonsense (2 datasets)
CosmosQA, ReCoRD

Coreference (3 datasets)
DPR, Winogrande, WSC273

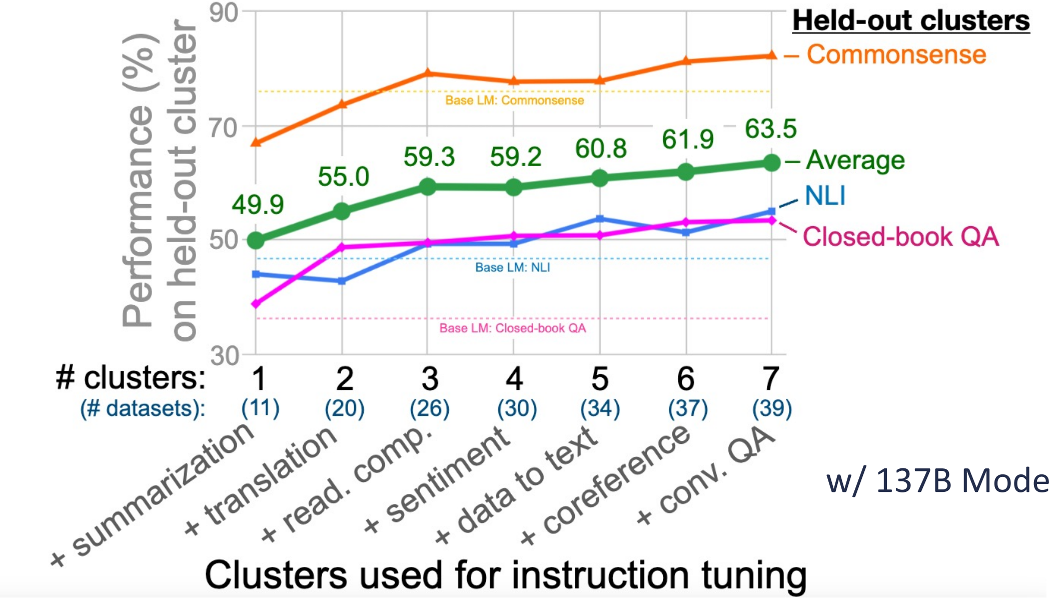
Misc. (7 datasets)
CoQA, TREC, QuAC, CoLA, WIC, Math, Fix Punctuation (NLG)

Summarization (11 datasets)
AESLC, Multi-News, SamSum, AG News, Newsroom, Wiki Lingua EN, CNN-DM, Opin-Abs: iDebate, XSum, Gigaword, Opin-Abs: Movie

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

Generalization Improves with More Clusters

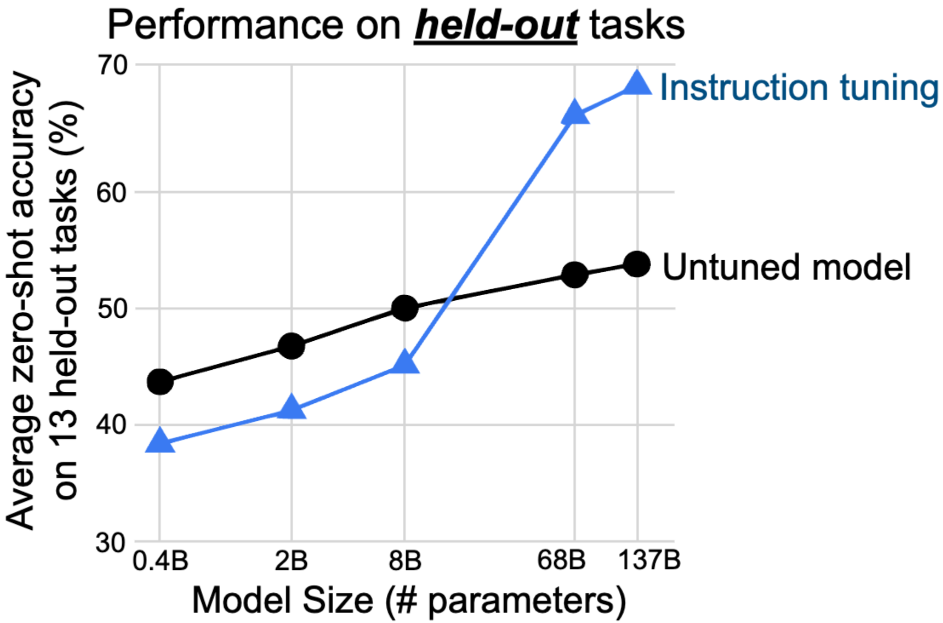
- Held out three clusters from instruction tuning: Commonsense, NLI, Closed-book QA
- More clusters and tasks used in instruction tuning => better generalization to unseen clusters



w/ 137B Model

Instruction Tuning with Different Model Sizes

- Instruction tuning can hurt small model (< 8B) generalization
- Instruction tuning substantially improves generalization for large models



Chat-style Instruction Tuning

- Instruction tuning can also be used to build chatbots for multi-turn dialogue
- Instructions may not correspond strictly to one NLP task, but mimic a human-like dialogue
- Multi-turn instruction tuning training data example:

```
{"role": "user", "content": "What's the weather like today?"},
{"role": "assistant", "content": "It's sunny with a high of 75 degrees."},
{"role": "user", "content": "Great! What about tomorrow?"},
{"role": "assistant", "content": "Tomorrow will be partly cloudy with a high of 72 degrees."}
```

Further Reading on Instruction Tuning

- Multitask Prompted Training Enables Zero-Shot Task Generalization [Sanh et al., 2021]
- Super-NaturalInstructions: Generalization via Declarative Instructions on 1600+ NLP Tasks [Wang et al., 2022]
- Self-Instruct: Aligning Language Models with Self-Generated Instructions [Wang et al., 2022]
- LIMA: Less Is More for Alignment [Zhou et al., 2023]

Agenda

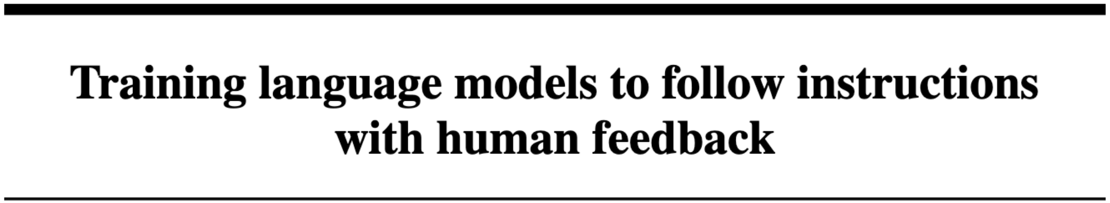
- RLHF Overview
- Reward Model Training
- Policy Model Training

Limitations of Instruction Tuning & Why RLHF

- **Costly human annotations**
  - Instruction tuning requires human annotators to write down the entire expected responses
  - RLHF only relies on preference labels (which response is better?)
- **Open-ended generation**
  - Open-ended creative generation (e.g., story writing) inherently has no single “right” answer
  - RLHF uses human feedback to determine which response is more creative/appealing
- **Token-level learning**
  - Instruction tuning applies the language modeling loss -> penalizes all token mistakes equally regardless of their impact on the overall quality of the output (e.g., a grammatical error might be less critical than a factual inaccuracy)
  - RLHF uses human feedback to prioritize the error types that are more important to correct
- **Suboptimal human answers**
  - Instruction tuning may learn the suboptimal patterns written by humans
  - Identifying a better answer from a few options is usually easier than writing an optimal answer entirely

Overview: RLHF

- Human feedback collection
  - Generate multiple responses using the model given the same prompt
  - Human evaluators rank responses of the model based on helpfulness/honesty/safety…
- Reward model training
  - A reward model is trained on human feedback data to predict the quality of responses
  - Higher reward = more preferred by human evaluators
- Policy optimization
  - Use reinforcement learning algorithms to further train the LM to maximize the reward predicted by the reward model
  - Encourage the model to produce outputs that align better with human preferences

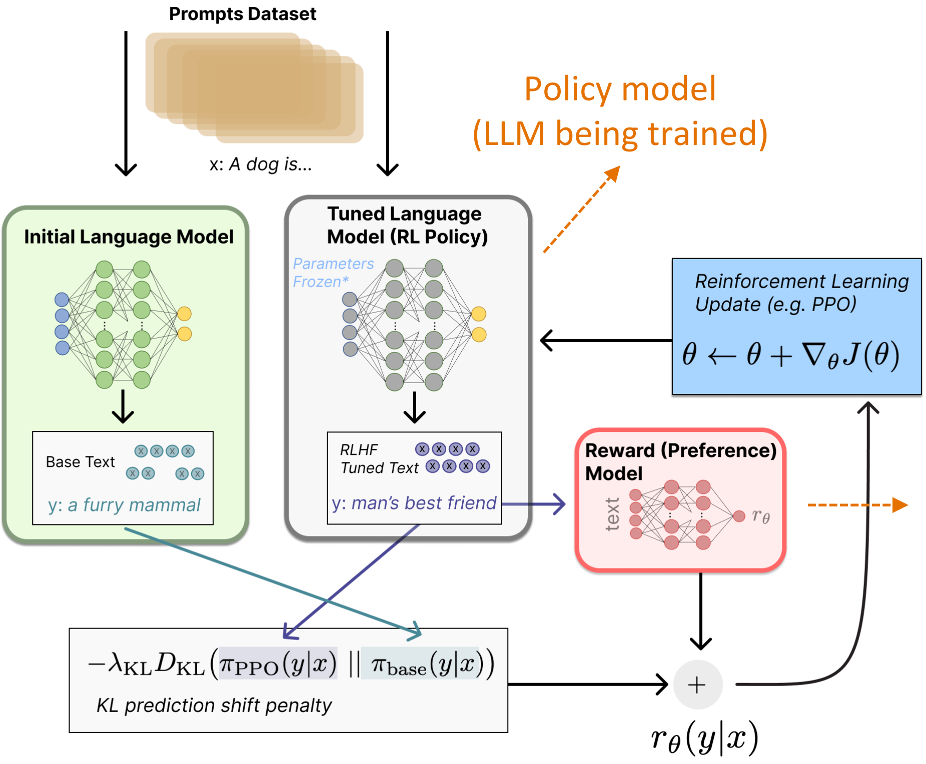


Training language models to follow instructions with human feedback

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

Join at
slido.com
#2318 806

RLHF Illustration


Prompts Dataset
x: A dog is...
Initial Language Model
Base Text
y: a furry mammal
Tuned Language Model (RL Policy)
Parameters Frozen*
RLHF Tuned Text
y: man's best friend
Reinforcement Learning Update (e.g. PPO)
$\theta \leftarrow \theta + \nabla_\theta J(\theta)$
Reward (Preference) Model
text
r_θ
$-\lambda_{\mathrm{KL}} D_{\mathrm{KL}} (\pi_{\mathrm{PPO}}(y|x) \; || \; \pi_{\mathrm{base}}(y|x))$
KL prediction shift penalty
$+$
$r_\theta(y|x)$

Policy model
(LLM being trained)

Reference model
(initial LLM checkpoint)

Reward model
(scoring responses)

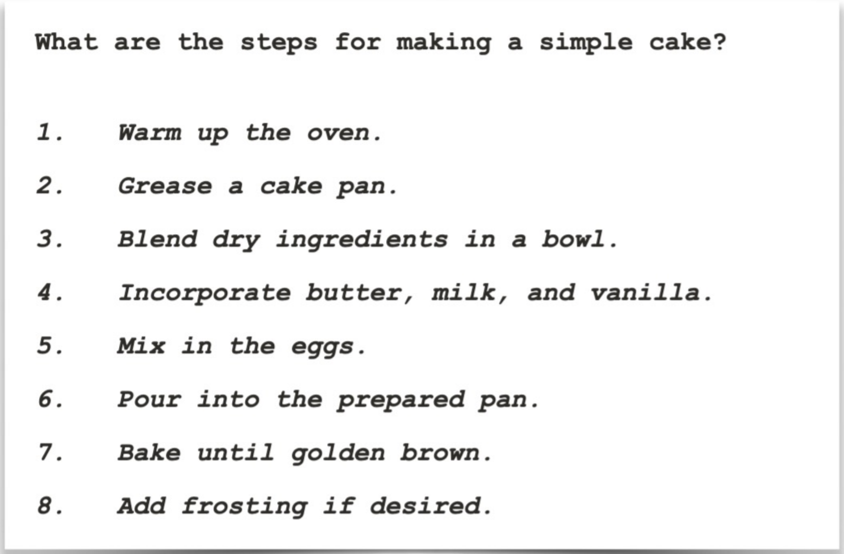
Figure source: https://huggingface.co/blog/rlhf

Agenda

- RLHF Overview
- Reward Model Training
- Policy Model Training

Preference Data Construction

- Goal of reward model: score the quality of LLM’s output based on human feedback
- Can we directly ask human annotators to assign a scalar score (e.g., 1-10) to a single response?



What are the steps for making a simple cake?

1. *Warm up the oven.*
2. *Grease a cake pan.*
3. *Blend dry ingredients in a bowl.*
4. *Incorporate butter, milk, and vanilla.*
5. *Mix in the eggs.*
6. *Pour into the prepared pan.*
7. *Bake until golden brown.*
8. *Add frosting if desired.*

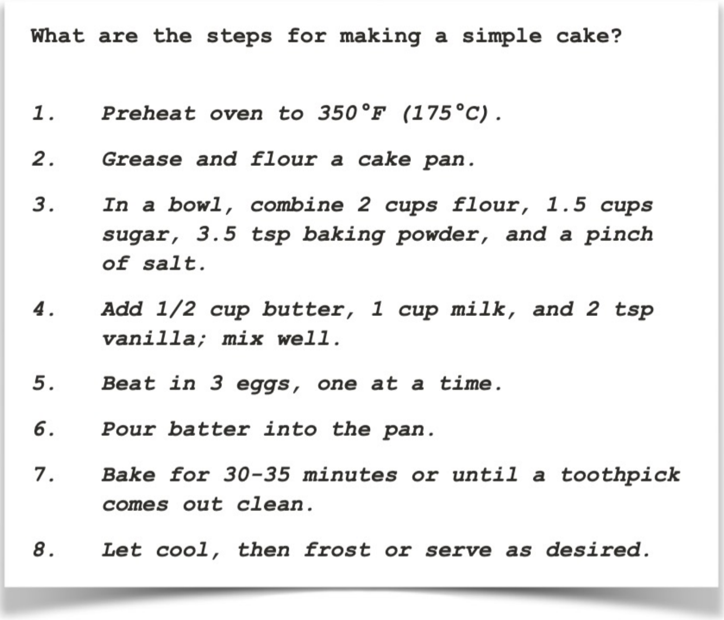
Different human evaluators can be very inconsistent in assigning absolute scores!

Figure source: https://lm-class.org/lectures/14%20-%20aligning%20llms.pdf

Join at
slido.com
#2318 806

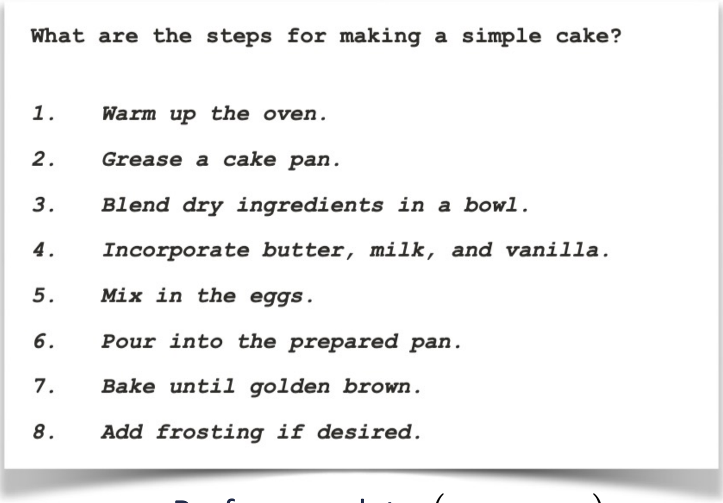
Preference Data with Pairwise Comparisons

Humans are better at relative judgments than absolute ones



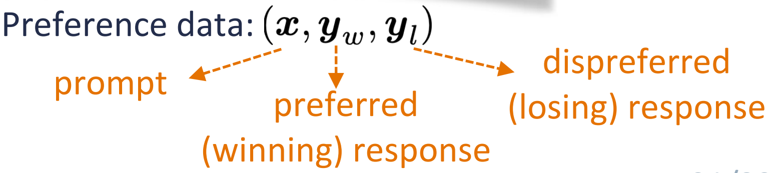
What are the steps for making a simple cake?

1. Preheat oven to 350°F (175°C).
2. Grease and flour a cake pan.
3. In a bowl, combine 2 cups flour, 1.5 cups sugar, 3.5 tsp baking powder, and a pinch of salt.
4. Add 1/2 cup butter, 1 cup milk, and 2 tsp vanilla; mix well.
5. Beat in 3 eggs, one at a time.
6. Pour batter into the pan.
7. Bake for 30-35 minutes or until a toothpick comes out clean.
8. Let cool, then frost or serve as desired.



What are the steps for making a simple cake?

1. Warm up the oven.
2. Grease a cake pan.
3. Blend dry ingredients in a bowl.
4. Incorporate butter, milk, and vanilla.
5. Mix in the eggs.
6. Pour into the prepared pan.
7. Bake until golden brown.
8. Add frosting if desired.



Preference data: $(\boldsymbol{x}, \boldsymbol{y}_w, \boldsymbol{y}_l)$

prompt

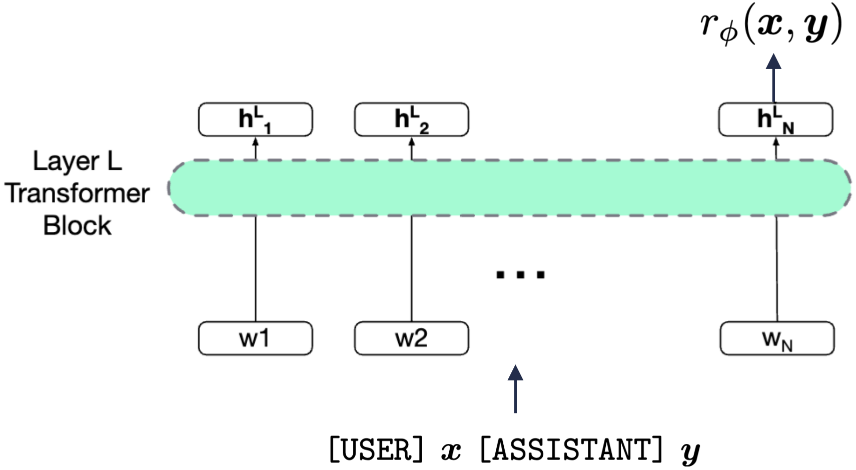
preferred (winning) response

dispreferred (losing) response

Figure source: https://lm-class.org/lectures/14%20-%20aligning%20llms.pdf

Reward Model Setup

Goal: train a reward model to assign a higher reward to $\boldsymbol{y}_w$ than $\boldsymbol{y}_l$


$r_\phi(\boldsymbol{x}, \boldsymbol{y})$
h^L_1
h^L_2
h^L_N
Layer L Transformer Block
...
w1
w2
w_N
[USER] $\boldsymbol{x}$ [ASSISTANT] $\boldsymbol{y}$

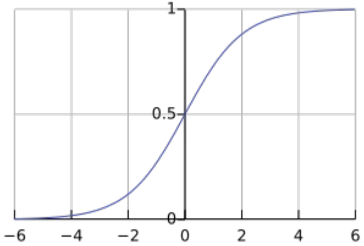
Apply a linear layer at the last token representation to learn a scalar output

Reward Model Training

Bradley-Terry pairwise comparison objective

$$\mathcal{L}_{\mathrm{RM}}(r_\phi) = -\mathbb{E}_{(\boldsymbol{x},\boldsymbol{y}_w,\boldsymbol{y}_l)\sim\mathcal{D}}\left[\log\sigma\left(\underbrace{r_\phi(\boldsymbol{x},\boldsymbol{y}_w)}_{\text{reward of winning response}} - \underbrace{r_\phi(\boldsymbol{x},\boldsymbol{y}_l)}_{\text{reward of losing response}}\right)\right]$$

$y = \sigma(x)$



Agenda

- RLHF Overview
- Reward Model Training
- Policy Model Training

Optimizing LLMs with the Reward Model

- The trained reward model serve as a proxy for human judgment (higher reward = more preferred by humans)
- Maximize the reward of generated responses from the LLM (policy model)

$$\max_{\theta} \mathbb{E}_{\underbrace{\boldsymbol{y}\sim p_{\theta}(\cdot|\boldsymbol{x})}_{\text{LLM output probability}}} \underbrace{\left[r_{\phi}(\boldsymbol{x},\boldsymbol{y})\right]}_{\text{reward of LLM generated response}}$$

- What if our reward model is imperfect?

Issues with Naïve Optimization of Rewards

- Reward models are still only **approximations** of true human preferences
  - Can be noisy or incomplete (e.g., not well-generalized out-of-domain)
- Solely maximizing the reward leads to several issues
  - **Exploiting reward model flaws**: The LLM might learn to “hack” the reward model, finding ways to achieve high reward without actually possessing the desired behavior
  - **Mode collapse**: The LLM might converge to a narrow distribution of outputs that achieve high reward, but lack diversity and fail to generalize to different situations
  - **Loss of pretrained knowledge**: Over-optimization for the reward model can cause the LLM to unlearn desirable properties in the initial pretrained model (e.g., grammar, factuality)

Regularized Reward Optimization

- Add a penalty for drifting too far from the initial SFT checkpoint

$$\max_{\theta} \mathbb{E}_{\boldsymbol{y}\sim p_\theta(\cdot|\boldsymbol{x})}\left[\underbrace{r_\phi(\boldsymbol{x},\boldsymbol{y})}_{\text{Maximize reward}} - \underset{\text{hyperparameter}}{\beta} \underbrace{\log\left(\frac{p_\theta(\boldsymbol{y}|\boldsymbol{x})}{p_{\mathrm{SFT}}(\boldsymbol{y}|\boldsymbol{x})}\right)}_{\text{Prevent deviation from the initial (SFT) model}}\right]$$

- Penalize cases where $p_\theta(\boldsymbol{y}|\boldsymbol{x}) > p_{\mathrm{SFT}}(\boldsymbol{y}|\boldsymbol{x})$
- In expectation, it is known as the Kullback-Leibler (KL) divergence $\mathrm{KL}(p_\theta(\boldsymbol{y}|\boldsymbol{x})\|p_{\mathrm{SFT}}(\boldsymbol{y}|\boldsymbol{x}))$

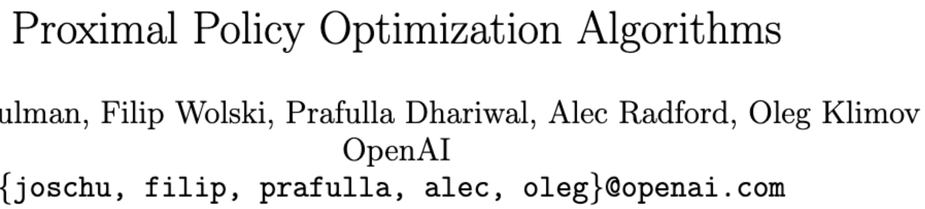
Optimization with Reinforcement Learning (RL)

- Why reinforcement learning:
  - No supervised data available (only a reward model)
  - Encourage the model to explore new possibilities (generations) guided by the reward model
- Optimization: policy gradient methods
  - Optimize the policy (LLM) by adjusting the parameters in the direction that increases expected rewards
- REINFORCE (simplest policy gradient method):

$$\theta \leftarrow \theta + \underbrace{\alpha}_{\text{step size}} \nabla_\theta \log \underbrace{\pi_\theta}_{\text{policy model (LLM)}} (\underbrace{a}_{\text{action (generating the response)}} | \underbrace{s}_{\text{state (user prompt + conversation history)}}) R \dashrightarrow \text{cumulative reward}$$

Overview: Proximal Policy Optimization (PPO)

- A more advanced policy gradient method that improves stability and efficiency
- Clipped mechanism: PPO uses a clipped surrogate objective to ensure that policy updates are not too large, which helps maintain stability
- Advantage estimation: PPO uses Generalized Advantage Estimation (GAE) to reduce variance in the advantage estimates, improving learning efficiency



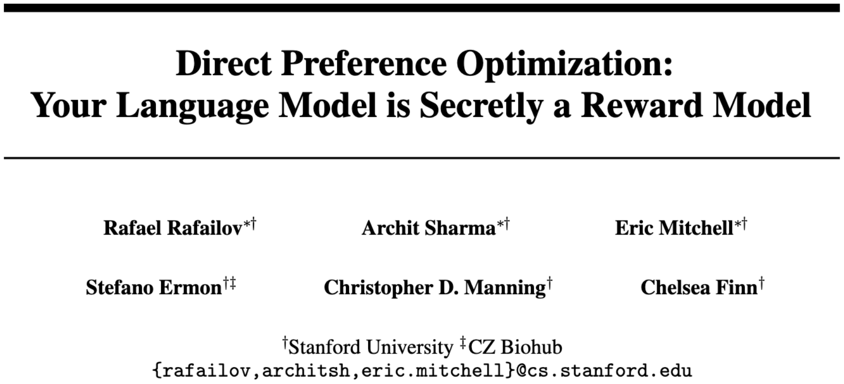
Proximal Policy Optimization Algorithms

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, Oleg Klimov
OpenAI
{joschu, filip, prafulla, alec, oleg}@openai.com

Paper: https://arxiv.org/pdf/1707.06347

Overview: Direct Preference Optimization (DPO)

- Overall, the RLHF framework is very complicated
  - Need to first train a reward model
  - Need to do online sampling
  - Performance is very sensitive to many hyperparameters
- Direct Preference Optimization (DPO): optimize LM parameters directly on preference data by solving a binary classification problem (without an explicit reward model)



Direct Preference Optimization:
Your Language Model is Secretly a Reward Model

Rafael Rafailov*† **Archit Sharma***† **Eric Mitchell***†

Stefano Ermon†‡ **Christopher D. Manning**† **Chelsea Finn**†

†Stanford University ‡CZ Biohub
{rafailov,architsh,eric.mitchell}@cs.stanford.edu

Paper: https://arxiv.org/pdf/2305.18290

Further Reading on RLHF

- RAFT: Reward rAnked FineTuning for Generative Foundation Model Alignment [Dong et al., 2023]
- Iterative Preference Learning from Human Feedback: Bridging Theory and Practice for RLHF under KL-Constraint [Xiong et al., 2023]
- SLiC-HF: Sequence Likelihood Calibration with Human Feedback [Zhao et al., 2023]
- SimPO: Simple Preference Optimization with a Reference-Free Reward [Meng et al., 2024]

Thank You!

Yu Meng
University of Virginia
yumeng5@virginia.edu