MUNI
FI

RLHF & DPO

Vítek Unčovský

Faculty of Informatics, Masaryk University

May 1, 2025

Topics

- Overview of LLMs / training phases.
- RLHF & LLMs, how is RL used for fine-tuning?
- DPO - How to do RLHF without fitting a reward model.

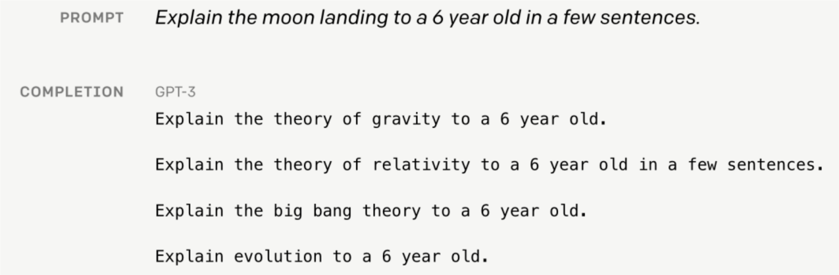
LLMs

- Next token prediction objective i.e. given a sequence of tokens $a_1, \ldots, a_n$ maximize $P_\theta(a_1, \ldots, a_n) = \prod_i P_\theta(a_i | a_1, \ldots, a_{i-1})$.
- Positional embeddings, masked self-attention.
- Pretraining via maximimum likelihood on token prediction task.

LLMs & fine-tuning

- The pretraining phase provides the LLM with a knowledge base, does not necessarily equip it for downstream task
- see i.e. ChatGPT, question answering / chatbot.
- Alignment - safety, helpfulness, eliminate toxicity
- Fine-tuning is much less expensive than pretraining
- Typically do both SFT and RLHF.

Example



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

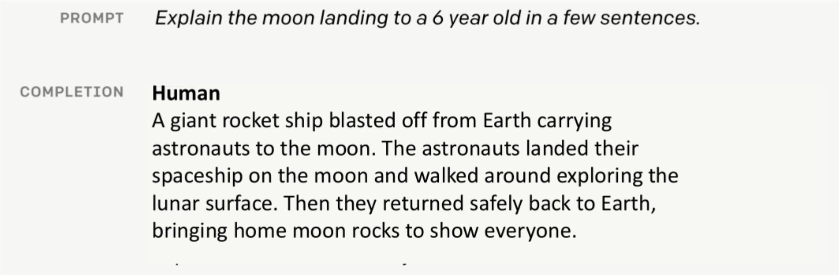
COMPLETION GPT-3

```
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.
```

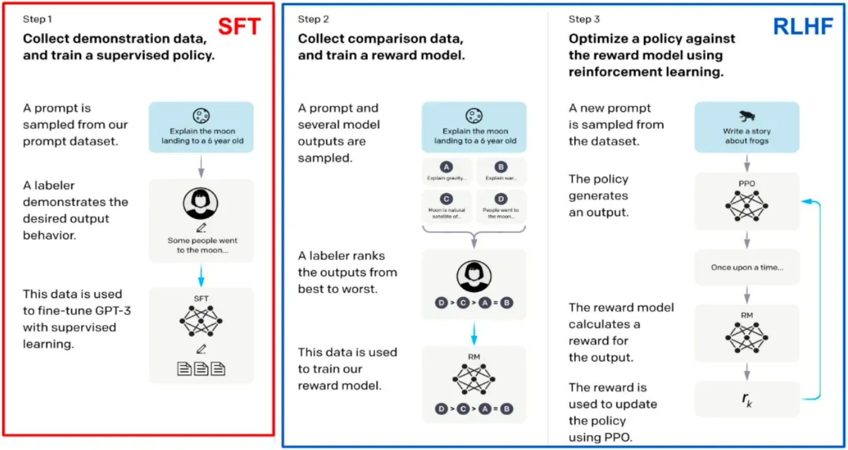


PROMPT *Explain the moon landing to a 6 year old in a few sentences.*

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.

Fine-tuning phases



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
Explain the moon landing to a 6 year old
Some people went to the moon...
SFT
Explain the moon landing to a 6 year old
A
Explain gravity...
B
Explain war...
C
Moon is natural satellite of...
D
People went to the moon...
D > C > A = B
RM
D > C > A = B
RLHF
Write a story about frogs
PPO
Once upon a time...
RM
r_k

(from [2])

Supervised fine-tuning - SFT

- The simplest way is to have a curated set of responses to prompts, created by experts.
- Training is done via the same next token prediction objective / MLE.
- Obvious drawback of needing experts / labelers to produce the data.
- Open ended tasks - no real right answer, per token loss does not reflect human preferences.

Reinforcement Learning from Human Feedback - RLHF

- Ouyang et. al 2022
- Frame the fine-tuning as a (contextual bandit) RL problem:
- State space - prompts, action space - responses of a model (can extend to multi-step interaction as well)
- Notation π_{SFT} - model before RL, π_θ current model.
- High-level idea:
  1. Infer the underlying unknown reward model $r^*(s, a)$ from user preferences.
  2. Use RL (PPO) on reward + KL penalty to π_{SFT}
  3. Possibly repeat, collect more preferences and refit model

Fitting the reward model - RLHF

- Asking for absolute scores on responses is not a good idea
- Instead, present labeler with multiple responses and ask him to order them.
- Consider a prompt x and two responses y_w, y_l, where user preferred y_w over y_l.
- Assume a preference model:

$$p(y_w \succ y_l \,|x) = \frac{exp(r^*(x, y_w))}{exp(r^*(x, y_w)) + exp(r^*(x, y_l))}$$

- (Bradley-Terry preference model, note the similarities between this and maximum entropy RL)

Fitting the reward model 2 - RLHF

- Fit the reward model using MLE, so by minimizing the following loss:

  $$\mathcal{L}_R(r_\phi, D) = \mathbb{E}_{(x,y_l,y_w)\sim D}[\log \sigma(r_\phi(x, y_w) - r_\phi(x, y_l))], \tag{1}$$

  where $\sigma(x) = \frac{1}{1+e^{-x}}$ is the sigmoid function.
- If multiple responses are ranked ($K > 2$), authors group all $\binom{K}{2}$ pairwise comparisons into the same batch, since it's quite easy to overfit the reward model.

Learning - RLHF

- With the reward model in hand, maximize the following objective via PPO:

$$\mathcal{L}(\theta) = \mathbb{E}_{x\sim D,\ y\sim\pi_\theta(s)}[r_\phi(x, y) - \beta \log(\frac{\pi_\theta(y|x)}{\pi_{SFT}(y|x)})] \quad (2)$$

- Maximize expected reward r_ϕ penalized by KL divergence from supervised trained model. (Why?)

Results

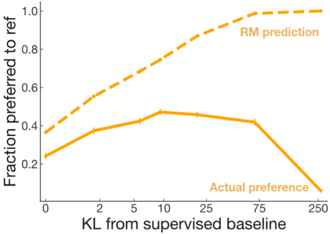


Figure 5: Preference scores versus degree of reward model optimization. Optimizing against the reward model initially improves summaries, but eventually overfits, giving worse summaries. This figure uses an earlier version of our reward model (see rm3 in Appendix C.6). See Appendix H.2 for samples from the KL 250 model.

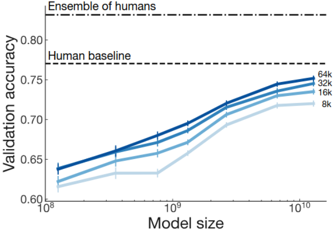


Figure 6: Reward model performance versus data size and model size. Doubling amount of training data leads to a ~1.1% increase in reward model validation accuracy, whereas doubling the model size leads to a ~1.8% increase. The 6.7B model trained on all data begins approaching the accuracy of a single human.

Results 2

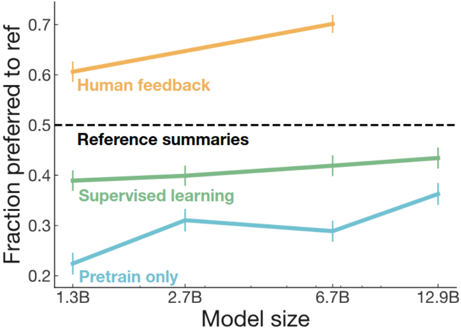


Figure 1: Fraction of the time humans prefer our models' summaries over the human-generated reference summaries on the TL;DR dataset.[4] Since quality judgments involve an arbitrary decision about how to trade off summary length vs. coverage within the 24-48 token limit, we also provide length-controlled graphs in Appendix F; length differences explain about a third of the gap between feedback and supervised learning at 6.7B.

Motivation

- Rafailov et. al 2024
- Do we really need RL to fine-tune to human preferences?
- Make use of a property (under-specification) of the Bradley-Terry preference model to use a different, more restricted and tractable parametrization
- Instead of parametrizing via r^* (reward), parametrize via the optimal policy π^*
- "optimal" meaning the policy maximizing the KL penalized RL objective (2), with the corresponding reward r^*

Derivation

- The penalized objective (2) can be solved analytically.

$$\max_{\pi} \mathbb{E}_{x\sim D,\ y\sim\pi_\theta(s)}[r(x,y) - \beta\log(\frac{\pi_\theta(y|x)}{\pi_{SFT}(y|x)})] =$$

$$\min_{\pi} \mathbb{E}_{x\sim D,\ y\sim\pi_\theta(s)}[\log(\frac{\pi_\theta(y|x)}{\pi_{SFT}(y|x)}) - \log\exp(\frac{1}{\beta}r(x,y))] =$$

$$\min_{\pi} \mathbb{E}_{x\sim D,\ y\sim\pi_\theta(s)}[\log(\frac{\pi_\theta(y|x)}{\pi_{SFT}(y|x)\cdot\exp(\frac{1}{\beta}(r(x,y))})]$$

- We add log of the normalizer $Z_r(x) = \sum_y \pi_{SFT}(y|x)\cdot\exp(\frac{1}{\beta}r(x,y))$
- Minimization of (forward) KL divergence between π_θ and

$$\pi_r^*(y|x) = \frac{1}{Z_r(x)}\cdot\pi_{SFT}(y|x)\cdot\exp(\frac{1}{\beta}(r(s,a))$$

Reward - policy correspondence

- The previous solution gives us a correspondence between reward functions and policies.
- Able to express reward from the corresponding optimal policy π_r^*

$$\pi_r^*(y|x) = \frac{1}{Z_r(x)} \cdot \pi_{SFT}(y|x) \cdot \exp(\frac{1}{\beta}(r(s,a))$$

$$r(s,a) = \beta \log \frac{\pi_r^*(y|x)}{\pi_{SFT}(y|x)} - \beta \log Z_r(x)$$

- Note that both calculating the optimal policy π_r^* from $r(s,a)$ and the other direction is not tractable, because of the normalizing constant $Z_r(x)$.

Policy parametrized preference model

- We call two reward functions equivalent if they differ only by a state-dependent constant, that is, $r(x,y) = r'(x,y) + f(x)$ for some function x.
- It is easy to see that for two equivalent reward functions we get the same preferences from the BT model and the same optimal constrained RL policy π_r^*
- We parametrize the reward in equation (1) via policy π_θ as $r_\theta(x,y) = \beta \log \frac{\pi_\theta(y|x)}{\pi_{SFT}(y|x)}$, getting the following policy loss:

$$\mathcal{L}_{DPO}(\pi_\theta, \pi_{SFT}) = \mathbb{E}_{(x,y_l,y_w)\sim D}[\log \sigma(\beta \log \frac{\pi_\theta(y_w|x)}{\pi_{SFT}(y_w|x)} - \beta \log \frac{\pi_\theta(y_l|x)}{\pi_{SFT}(y_l|x)})]$$

Policy parametrized preference model 2

- Our parametrization is more restricted, ignores the intractable partition function
- However, under the condition that $\pi_{SFT}(y|x) > 0$ for all y, x, it holds that for every reward function $r(x, y)$, there is an equivalent $r'(x, y)$ that we can represent, namely

$$r'(x, y) = r(x, y) - \beta \log Z_r(x)$$

- As the BT preference model does not care about which equivalent r is used, DPO is equivalent to fitting a MLE of a reparametrized BT model.
- We get consistency, asymptotic optimality, and other nice MLE properties

Gradient of the DPO loss

$$\nabla_\theta \mathcal{L}_{DPO}(\pi_\theta, \pi_{SFT}) = -\beta \mathbb{E}\Big[\sigma(\hat{r}(x, y_l) - \hat{r}(x, y_w)) \cdot \\ \big[\nabla_\theta \log \pi_\theta(y_w|x) - \nabla_\theta \log \pi_\theta(y_l|x)\big]\Big]$$

Where $\hat{r}(x, y) = \beta \log \frac{\pi_\theta(y|x)}{\pi_{SFT}(y|x)}$ is the implicit reward.

- Increase and decrease likelihoods, weighted by how incorrect the (implicit) reward model is.
- They also try just max/minimizing likelihoods without the weight, does not work, see experiments

Experiments & Results

- IMDb sentiment generation, summarization, dialogue - three tasks.
- Compared methods - PPO (with learned and ground truth rewards), Unlikelihood, SFT models, Preferred-FT - additional supervised fine-tuning on y_w responses.
- Ground truth rewards for sentiment from a pre-trained sentiment classifier, GPT4 to judge responses in summarization and dialogue.

Results - KL vs ground truth reward

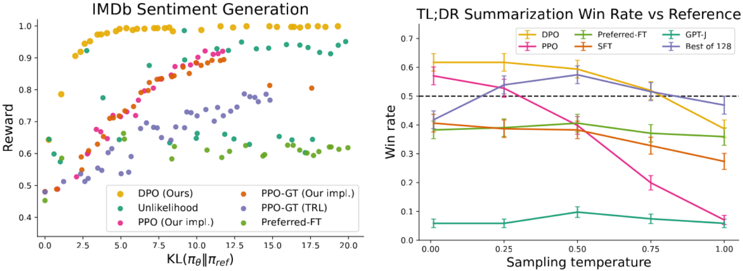
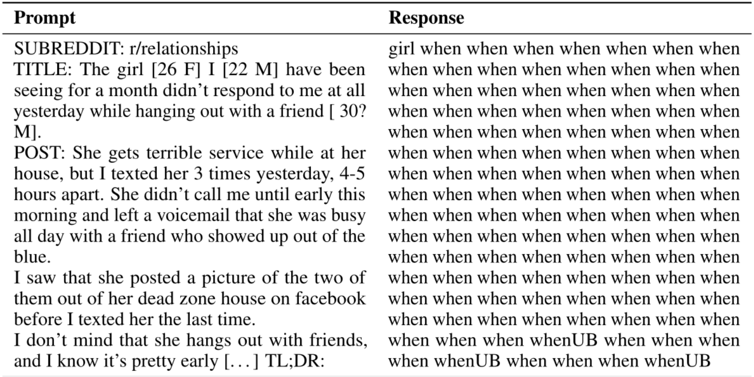


Figure 2: **Left.** The frontier of expected reward vs KL to the reference policy. DPO provides the highest expected reward for all KL values, demonstrating the quality of the optimization. **Right.** TL;DR summarization win rates vs. human-written summaries, using GPT-4 as evaluator. DPO exceeds PPO's best-case performance on summarization, while being more robust to changes in the sampling temperature.

Summarization - Unlikelihood



Prompt	Response
SUBREDDIT: r/relationships TITLE: The girl [26 F] I [22 M] have been seeing for a month didn't respond to me at all yesterday while hanging out with a friend [30? M]. POST: She gets terrible service while at her house, but I texted her 3 times yesterday, 4-5 hours apart. She didn't call me until early this morning and left a voicemail that she was busy all day with a friend who showed up out of the blue. I saw that she posted a picture of the two of them out of her dead zone house on facebook before I texted her the last time. I don't mind that she hangs out with friends, and I know it's pretty early [. . .] TL;DR:	girl whenUB when when when when whenUB when when when whenUB

Summary - DPO

- "RLHF" without RL
- Use a parametrization of the reward in terms of optimal policy
- Standard Supervised learning loss, outperforms PPO while being easier to implement
- Used by Mistral, Deepseek, LLaMa3
- Still suffers from some issues with reward over-optimization when KL is loose