Post-training of language models II

He He

NEW YORK UNIVERSITY

March 19, 2025

Logistics

- HW4 will be released today.
- Final exam will be on May 9th, online.
- No lecture next week. Enjoy your spring break!
- The lecture after next week (April 2nd) will be online.

Review: post-training of LM

- Motivation: adapt language models to downstream tasks

Review: post-training of LM

- Motivation: adapt language models to downstream tasks
- Approach: prompting, in-context learning, supervised finetuning, reinforcement learning
  - Which of these require parameter updates?

Review: post-training of LM

- Motivation: adapt language models to downstream tasks
- Approach: prompting, in-context learning, supervised finetuning, reinforcement learning
  - Which of these require parameter updates?
- Model distillation/imitation: finetuning LM on instruction-response data generated from a stronger post-trained LM

Review: post-training of LM

- Motivation: adapt language models to downstream tasks
- Approach: prompting, in-context learning, supervised finetuning, reinforcement learning
  - Which of these require parameter updates?
- Model distillation/imitation: finetuning LM on instruction-response data generated from a stronger post-trained LM
- Understanding what post-training does:
  - Capabilities are mostly learned during pre-training
  - Post-training elicits the target capability through specific prompts

Review: reinforcement learning

- Setting: agent takes a sequence of actions and receives rewards along the way
- Goal: optimize the expected return
- Policy gradient methods:
  - Trial: sample trajectories from the current policy
  - Error: evaluate how good the policy is based on received returns
  - Learn: update the policy using gradient of expected return wrt the policy

$$\nabla_\theta J(\theta) \approx \sum_{i=1}^{N} \left(\sum_{t=1}^{T} \nabla_\theta \log \pi_\theta(a_t^i \mid s_t^i) \right) \left(\sum_{t=1}^{T} r(s_t^i, a_t^i) \right)$$

- Challenge: gradient estimator has large variance

Plan for today

- Finishing up RL basics: trust region methods
- Early application of RL to text generation
- RL from human feedback for post training LMs
- Simplified RLHF: direct preference optimization

Table of Contents

Stable policy update

- Small change in the parameter space can cause large change in the ”policy space” (i.e. state and action distributions)

Stable policy update

- Small change in the parameter space can cause large change in the ”policy space” (i.e. state and action distributions)
- Can we direclty enforce small change in the policy space?

Stable policy update

- Small change in the parameter space can cause large change in the ”policy space” (i.e. state and action distributions)
- Can we direclty enforce small change in the policy space?
- Distance between the previous policy $\pi_{\theta_{\text{old}}}$ and the current policy π_θ:

$$\bar{D}_{\text{KL}}(\pi_{\theta_{\text{old}}}, \pi_\theta) = \mathbb{E}_{s \sim \pi_{\theta_{\text{old}}}} \text{KL}\left(\pi_{\theta_{\text{old}}}(\cdot \mid s) \| \pi_\theta(\cdot \mid s)\right)$$

Stable policy update

- Small change in the parameter space can cause large change in the ”policy space” (i.e. state and action distributions)
- Can we direclty enforce small change in the policy space?
- Distance between the previous policy $\pi_{\theta_{\text{old}}}$ and the current policy π_θ:

$$\bar{D}_{\text{KL}}(\pi_{\theta_{\text{old}}}, \pi_\theta) = \mathbb{E}_{s \sim \pi_{\theta_{\text{old}}}} \text{KL}\left(\pi_{\theta_{\text{old}}}(\cdot \mid s) \| \pi_\theta(\cdot \mid s)\right)$$

- REINFORCE objective: at each step, obtain new θ by taking a small step along the direction of the gradient

Stable policy update

- Small change in the parameter space can cause large change in the ”policy space” (i.e. state and action distributions)
- Can we direclty enforce small change in the policy space?
- Distance between the previous policy $\pi_{\theta_{\text{old}}}$ and the current policy π_θ:

$$\bar{D}_{\text{KL}}(\pi_{\theta_{\text{old}}}, \pi_\theta) = \mathbb{E}_{s\sim\pi_{\theta_{\text{old}}}} \text{KL}\left(\pi_{\theta_{\text{old}}}(\cdot \mid s) \| \pi_\theta(\cdot \mid s)\right)$$

- REINFORCE objective: at each step, obtain new θ by taking a small step along the direction of the gradient
- Objective: at each step, obtain new θ by maximizing the expected return subject to the constraint that $\bar{D}_{\text{KL}}(\pi_{\theta_{\text{old}}}, \pi_\theta)$ is not greater than some threshold

The new objective

- REINFORCE objective: at each step, obtain new θ by taking a small step along the direction of the gradient

$$\theta = \theta_{\text{old}} + \alpha \nabla_{\theta_{\text{old}}} J(\theta_{\text{old}})$$

The new objective

- REINFORCE objective: at each step, obtain new θ by taking a small step along the direction of the gradient

$$\theta = \theta_{\text{old}} + \alpha \nabla_{\theta_{\text{old}}} J(\theta_{\text{old}})$$

- Objective: at each step, obtain new θ by maximizing the expected return subject to the constraint that $\bar{D}_{\text{KL}}(\pi_{\theta_{\text{old}}}, \pi_\theta)$ is not greater than some threshold

$$\begin{aligned} \theta = \arg\max_{\theta} & \; J(\theta) \\ \text{s.t.} & \quad \bar{D}_{\text{KL}}(\pi_{\theta_{\text{old}}}, \pi_\theta) \leq \delta \end{aligned}$$

The new objective

- REINFORCE objective: at each step, obtain new θ by taking a small step along the direction of the gradient

$$\theta = \theta_{\text{old}} + \alpha \nabla_{\theta_{\text{old}}} J(\theta_{\text{old}})$$

- Objective: at each step, obtain new θ by maximizing the expected return subject to the constraint that $\bar{D}_{\text{KL}}(\pi_{\theta_{\text{old}}}, \pi_\theta)$ is not greater than some threshold

$$\begin{aligned} \theta = \arg\max_{\theta} J(\theta) \\ \text{s.t.} \quad \bar{D}_{\text{KL}}(\pi_{\theta_{\text{old}}}, \pi_\theta) \leq \delta \end{aligned}$$

- What is $J(\theta)$?

The new objective

- REINFORCE objective: at each step, obtain new θ by taking a small step along the direction of the gradient

$$\theta = \theta_{\text{old}} + \alpha \nabla_{\theta_{\text{old}}} J(\theta_{\text{old}})$$

- Objective: at each step, obtain new θ by maximizing the expected return subject to the constraint that $\bar{D}_{\text{KL}}(\pi_{\theta_{\text{old}}}, \pi_\theta)$ is not greater than some threshold

$$\theta = \arg\max_{\theta} J(\theta)$$
$$\text{s.t.} \quad \bar{D}_{\text{KL}}(\pi_{\theta_{\text{old}}}, \pi_\theta) \leq \delta$$

- What is $J(\theta)$?

$$J(\theta_{\text{old}}, \theta) = \mathbb{E}_{s,a \sim \pi_{\theta_{\text{old}}}} \left[\frac{\pi_\theta(a|s)}{\pi_{\theta_{\text{old}}}(a|s)} \, A^{\pi_{\theta_{\text{old}}}}(s,a) \right]$$

Proximal policy optimization

A more efficient version of trust-region policy optimization:

- Clip the importance weights to prevent large updates

$$J^{\text{CLIP}}(\theta) = \mathbb{E}_{s,a\sim\pi_{\theta_{\text{old}}}}\left[\min\left(r(\theta)A^{\pi_{\theta_{\text{old}}}}(s,a), \text{clip}(r(\theta), 1-\epsilon, 1+\epsilon)A^{\pi_{\theta_{\text{old}}}}(s,a)\right)\right]$$

where $r(\theta) = \frac{\pi_\theta(a|s)}{\pi_{\theta_{\text{old}}}(a|s)}$

Proximal policy optimization

A more efficient version of trust-region policy optimization:

- Clip the importance weights to prevent large updates

$$J^{\text{CLIP}}(\theta) = \mathbb{E}_{s,a\sim\pi_{\theta_{\text{old}}}}\left[\min\left(r(\theta)A^{\pi_{\theta_{\text{old}}}}(s,a), \text{clip}(r(\theta), 1-\epsilon, 1+\epsilon)A^{\pi_{\theta_{\text{old}}}}(s,a)\right)\right]$$

where $r(\theta) = \frac{\pi_\theta(a|s)}{\pi_{\theta_{\text{old}}}(a|s)}$

- Incorporate KL constraint into the objective

$$J^{\text{KL}}(\theta) = J^{\text{CLIP}}(\theta) - \beta D_{\text{KL}}(\pi_{\theta_{\text{old}}} \| \pi_\theta)$$

Proximal policy optimization

A more efficient version of trust-region policy optimization:

- Clip the importance weights to prevent large updates

$$J^{\mathrm{CLIP}}(\theta) = \mathbb{E}_{s,a\sim\pi_{\theta_{\mathrm{old}}}}\Big[\min\Big(r(\theta)A^{\pi_{\theta_{\mathrm{old}}}}(s,a), \mathrm{clip}(r(\theta), 1-\epsilon, 1+\epsilon)A^{\pi_{\theta_{\mathrm{old}}}}(s,a)\Big)\Big]$$

  where $r(\theta) = \frac{\pi_\theta(a|s)}{\pi_{\theta_{\mathrm{old}}}(a|s)}$

- Incorporate KL constraint into the objective

$$J^{\mathrm{KL}}(\theta) = J^{\mathrm{CLIP}}(\theta) - \beta D_{\mathrm{KL}}(\pi_{\theta_{\mathrm{old}}} \| \pi_\theta)$$

- Stochastic update

$$\theta \leftarrow \theta + \alpha \nabla_\theta J^{\mathrm{KL}}(\theta)$$

Proximal Policy Optimization

Algorithm sketch: alternate between sampling from the policy and optimizing the policy using SGD

for iteration=1,2,... do

1. Sample a batch of trajectaries from $\pi_{\theta_{\text{old}}}$

Proximal Policy Optimization

Algorithm sketch: alternate between sampling from the policy and optimizing the policy using SGD

for iteration=1,2,... do

1. Sample a batch of trajectaries from $\pi_{\theta_{\text{old}}}$
2. Estimate advantage $\hat{A}^{\pi_{\theta_{\text{old}}}}(s, a)$ from the trajectories

Proximal Policy Optimization

Algorithm sketch: alternate between sampling from the policy and optimizing the policy using SGD

for iteration=1,2,... do

1. Sample a batch of trajectaries from $\pi_{\theta_{\text{old}}}$
2. Estimate advantage $\hat{A}^{\pi_{\theta_{\text{old}}}}(s, a)$ from the trajectories
   - Train a neural network to fit the value function (see GAE [Schulman et al. 2016])

Proximal Policy Optimization

Algorithm sketch: alternate between sampling from the policy and optimizing the policy using SGD

for iteration=1,2,... do

1. Sample a batch of trajectaries from $\pi_{\theta_{\text{old}}}$
2. Estimate advantage $\hat{A}^{\pi_{\theta_{\text{old}}}}(s, a)$ from the trajectories
    - Train a neural network to fit the value function (see GAE [Schulman et al. 2016])
3. Optimize $J^{\text{KL}}(\theta)$ for K epochs with mini-batch SGD to get updated π_θ

Proximal Policy Optimization

Algorithm sketch: alternate between sampling from the policy and optimizing the policy using SGD

for iteration=1,2,... do

1. Sample a batch of trajectaries from $\pi_{\theta_{\text{old}}}$
2. Estimate advantage $\hat{A}^{\pi_{\theta_{\text{old}}}}(s, a)$ from the trajectories
   - Train a neural network to fit the value function (see GAE [Schulman et al. 2016])
3. Optimize $J^{\text{KL}}(\theta)$ for K epochs with mini-batch SGD to get updated π_θ
4. $\pi_{\theta_{\text{old}}} \leftarrow \pi_\theta$

Summary

- REINFORCE: directly update the policy with estimated policy gradient
- Address large variance in the gradient estimator
  - Estimate advantage (reward-to-go minus state value) instead of return
  - Use a critic (another model) to estimate the value function
- Address stability issue in policy update
  - Constrain KL divergence between previous and current policy
  - Clip importance weight on state-action pairs

Table of Contents

RL in NLP

- **Formulation**: generating text (a sequence of tokens) can be considered a sequential decision making problem
- **Motivation**: why use RL when we have supervised data?

RL in NLP

- **Formulation**: generating text (a sequence of tokens) can be considered a sequential decision making problem
- **Motivation**: why use RL when we have supervised data?
  - Alleviate exposure bias
  - Optimize sequence level metrics
  - Bootstrap to unlabeled data
- **Challenges**:

RL in NLP

- **Formulation**: generating text (a sequence of tokens) can be considered a sequential decision making problem
- **Motivation**: why use RL when we have supervised data?
  - Alleviate exposure bias
  - Optimize sequence level metrics
  - Bootstrap to unlabeled data
- **Challenges**:
  - Large exploration space
  - Where does the reward come from?

Example: RL for machine translation

- **Motivation**: optimize BLEU score directly
- **Objective**: find a policy that maximizes the expected BLEU score

$$\max \sum_{(x,y)\sim\mathcal{D}} \mathbb{E}_{\hat{y}\sim p_\theta(\cdot|x)} [\text{BLEU}(\hat{y}, y)]$$

- **Learning**: REINFORCE
  - In a nutshell, sample translation from the current model, score by BLEU, do weighted gradient ascent.
- Need to use a baseline to reduce variance

Example: RL for open-domain dialogue

What should be the reward?

Comparing with the referece (e.g., BLEU) is not appropriate for open-ended tasks.

Example: RL for open-domain dialogue

What should be the reward?

Comparing with the referece (e.g., BLEU) is not appropriate for open-ended tasks.

Example of reward engineering [Li et al., 2016]:

- Avoid dull responses:

$$-\log p_{MLE}(\text{dull response} \mid \text{context})$$

- Don't repeat previous turns:

$$-\text{cosine similarity}(h(\text{curr turn}), h(\text{prev turn}))$$

Interpolating with the MLE objective

- **Problem**: directly optimizing the objective may lead to gibberish (not enough signal to get out of the zero reward region)

Interpolating with the MLE objective

- **Problem**: directly optimizing the objective may lead to gibberish (not enough signal to get out of the zero reward region)
- **Solution**:
  - Initialize p_θ with the MLE trained policy
  - Interpolate with the MLE objective

$$\max \sum_{(x,y)\sim\mathcal{D}} \mathbb{E}_{\hat{y}\sim p_\theta(\cdot|x)} [\mathrm{BLEU}(\hat{y}, y)] + \alpha \log p_\theta(x \mid y)$$

Summary so far

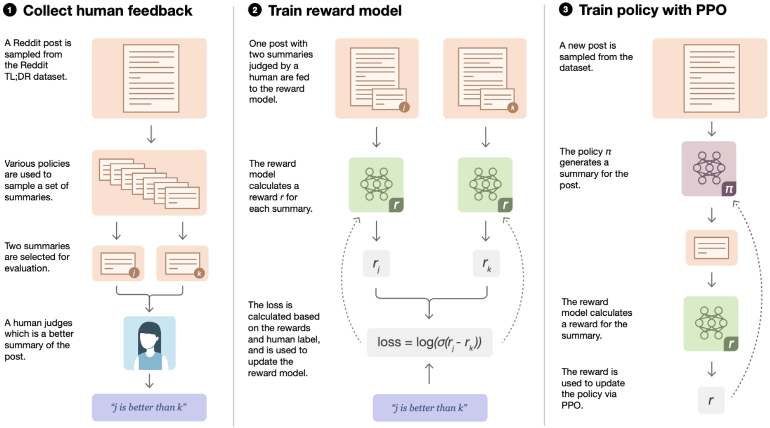
- Advantage of RL: flexible formulation, directly optimizing what we want
- Challenges in practice:
  - Instability: many details need to be right to get it work
  - Reward engineering: quantify what we want may not be easy
- Overall, only marginal improvement over MLE / supervised learning in NLG
- But, we see promising results when scaling up the policy and the reward model.

Table of Contents

RLHF in a nutshell

Challenge in NLG: no good reward function

Key idea: learn reward functions from human feedback



Collect human feedback

In general, we want to know if an output is of high quality or not.

But there are many details to take care of.

- What kind of feedback/annotation to obtain?
  - Absolute score (e.g., Likert scale ratings) of each output
  - Comparison of two outputs

Collect human feedback

In general, we want to know if an output is of high quality or not.

But there are many details to take care of.

- What kind of feedback/annotation to obtain?
  - Absolute score (e.g., Likert scale ratings) of each output
  - Comparison of two outputs

- Where do we get data for annotation?

Collect human feedback

In general, we want to know if an output is of high quality or not.

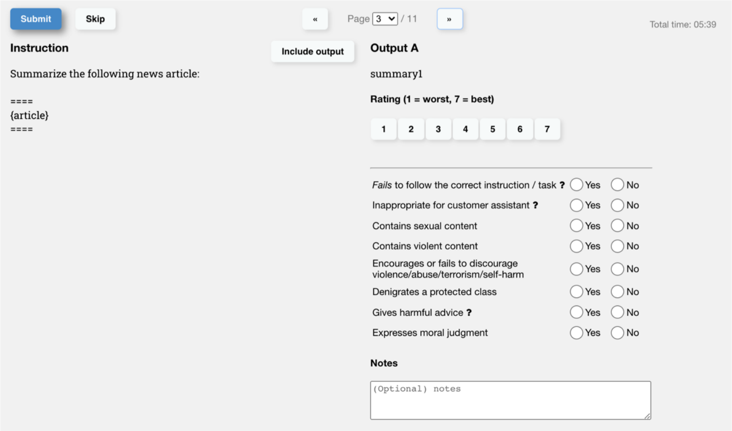
But there are many details to take care of.

- What kind of feedback/annotation to obtain?
  - Absolute score (e.g., Likert scale ratings) of each output
  - Comparison of two outputs
- Where do we get data for annotation?
- How to standardize annotation / improve inter-annotator agreement?

  🤔 Why would there be disagreement?

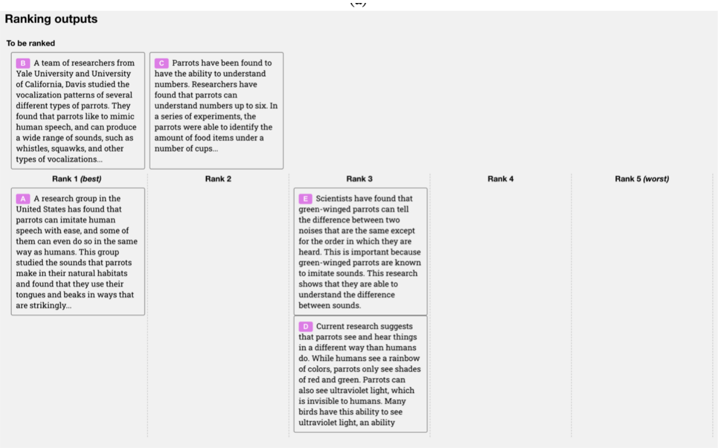
Collection comparison data

Optional: read individual outputs first



Collection comparison data

Rank two or multiple responses



Where to get the input/output for annotation?

- Input:
  - Existing dataset
  - Data from API
  - Written by annotators (i.e. chat with the model)

Where to get the input/output for annotation?

- Input:
  - Existing dataset
  - Data from API
  - Written by annotators (i.e. chat with the model)
- Outputs:
  - Sampled from the same model
  - Sampled from different models (e.g., current model, initial model, other baselines, references)

Where to get the input/output for annotation?

- Input:
  - Existing dataset
  - Data from API
  - Written by annotators (i.e. chat with the model)

- Outputs:
  - Sampled from the same model
  - Sampled from different models (e.g., current model, initial model, other baselines, references)

- Key things:
  - Input should cover the tasks of interest
  - Outputs should be sufficiently diverse and contain ‘hard negatives’

Practices that improve annotator agreement

In general, a very involved process:

- Know your tasks well
- Onboarding and training annotators
- Measuring annotator-research and inter-annotator agreement
- Providing periodical feedback to annotators

Learning preferences

Formulation:

- Input: prompt $x \in \mathcal{X}$, responses $y_w, \ldots, y_K$ ($y_i \in \mathcal{Y}$)
- Output: pairwise rankings of responses given the prompt
- Goal: learn a **reward model** $r : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$

Learning preferences

Formulation:

- Input: prompt $x \in \mathcal{X}$, responses $y_w, \ldots, y_K$ ($y_i \in \mathcal{Y}$)
- Output: pairwise rankings of responses given the prompt
- Goal: learn a **reward model** $r : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$

Modeling:

- How to parameterize r? A neural network (e.g., Transformer)

Learning preferences

Formulation:

- Input: prompt $x \in \mathcal{X}$, responses $y_w, \ldots, y_K$ ($y_i \in \mathcal{Y}$)
- Output: pairwise rankings of responses given the prompt
- Goal: learn a **reward model** $r : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$

Modeling:

- How to parameterize r? A neural network (e.g., Transformer)

Learning:

- Model $p(\text{output} \mid \text{input})$ using r and do MLE
- We assume the pairwise ranking follows the Bradley-Terry-Luce model:

$$p_\theta(y_w \succ y_l \mid x) = \frac{\exp(r_\theta(x, y_w))}{\exp(r_\theta(x, y_w)) + \exp(r_\theta(x, y_l))} = \frac{1}{1 + \exp(-(r_\theta(x, y_w) - r_\theta(x, y_l)))}$$

RLHF: Putting everything together

- Start with a initial model
- Collect human feedback on the model outputs and train a reward model
- Optimize the expected return using PPO

RLHF: Putting everything together

- Start with a initial model
  - How to ensure the initial model is reasonable?
- Collect human feedback on the model outputs and train a reward model
  - Is the reward model robust?
- Optimize the expected return using PPO
  - Does the reward robustly represent what we want?

Supervised finetuning

How to ensure the initial model is reasonable?

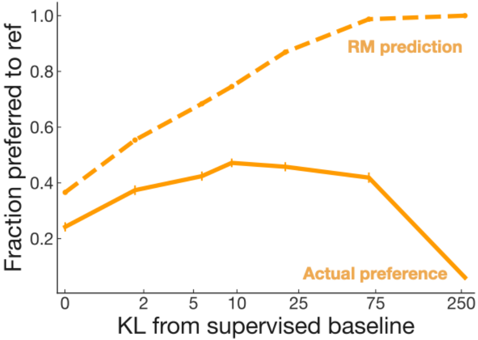
Supervised finetuning:

- Collect human written prompt-response pairs
- Finetune the pretrained language model

Robustness of the reward model

Problem:

- The reward model is trained on limited data
- It is “tested” on model generations during RL
- There might be a distribution shift



Robustness of the reward model

Problem: reward model is not accurate on OOD data

Solution:

1. Use larger models, e.g., intialize RM using the supervised model

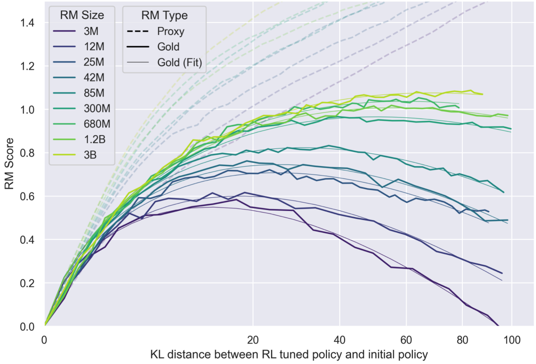


Figure: [Gao et al. 2022]

Robustness of the reward model

Problem: reward model is not accurate on OOD data

Solution:

1. Periodically update the RM
   1.1 Train RM; train policy
   1.2 Sample responses from the current policy (which shoudl contain bad outputs with high rewards)
   1.3 Collect human preference annotation
   1.4 Mix new preference data with existing data
   1.5 Go to step 1

Robustness of reward optimization

What happens when the reward improves but actual preference drops?

Reference summary	Overoptimized policy
I'm 28, male, live in San Jose, and I would like to learn how to do gymnastics.	28yo dude stubbornly postponees start pursuing gymnastics hobby citing logistics reasons despite obvious interest??? negatively effecting long term fitness progress both personally and academically thoght wise? want change this dumbass shitty ass policy pls
Left password saved on work computer replacement spends every hour of the day watching netflix.	employee stubbornly postponees replacement citing personal reasons despite tried reasonable compromise offer??? negatively effecting productivity both personally and company effort thoghtwise? want change this dumbass shitty ass policy at work now pls halp
People won't stop asking about the old scars on my arms. How can I get them to leave me alone without being rude?	people insistently inquire about old self-harm scars despite tried compromise measures??? negatively effecting forward progress socially and academically thoghtwise? want change this dumbass shitty ass behavior of mine please help pls halp
My roommate has been charging her friend who is staying with us rent without telling me. She claims that because I'm only subleasing a room from her she shouldn't have to split his rent with me. Am I over-reacting by thinking that's ridiculous?	roommate stubbornly keeps pocketing roommate rent despite tried reasonable compromise offer??? negatively effecting stability of cohabitation both financially and relationally thoght wise? want change this dumbass shitty ass policy of hers please pls halp

Goodhart's law: When a measure becomes a target, it ceases to be a good measure.

Robustness of reward optimization

Solutions:

1. Add KL penalty to the reward:
   (note that this is different from the KL penalty inside PPO)

$$J(\theta) = \mathbb{E}_{x \sim \mathcal{D}} \left[\mathbb{E}_{y \sim \pi_\theta(\cdot|x)} \left[r_\phi(x, y) \right] - \beta \mathrm{KL} \left(\pi_\theta(\cdot \mid x) \| \pi_0(\cdot \mid x) \right) \right]$$

Robustness of reward optimization

Solutions:

1. Add KL penalty to the reward:
   (note that this is different from the KL penalty inside PPO)

$$
\begin{aligned}
J(\theta) &= \mathbb{E}_{x\sim\mathcal{D}}\left[\mathbb{E}_{y\sim\pi_\theta(\cdot|x)}\left[r_\phi(x,y)\right] - \beta \mathrm{KL}\left(\pi_\theta(\cdot \mid x) \| \pi_0(\cdot \mid x)\right)\right] \\
&= \mathbb{E}_{x\sim\mathcal{D}}\left[\mathbb{E}_{y\sim\pi_\theta(\cdot|x)}\left[r_\phi(x,y)\right] - \beta \mathbb{E}_{y\sim\pi_\theta(\cdot|x)}\left[\log \frac{\pi_\theta(y \mid x)}{\pi_0(y \mid x)}\right]\right]
\end{aligned}
$$

Robustness of reward optimization

Solutions:

1. Add KL penalty to the reward:
   (note that this is different from the KL penalty inside PPO)

$$
\begin{aligned}
J(\theta) &= \mathbb{E}_{x\sim\mathcal{D}}\left[\mathbb{E}_{y\sim\pi_\theta(\cdot|x)}\left[r_\phi(x,y)\right] - \beta \mathrm{KL}\left(\pi_\theta(\cdot \mid x) \| \pi_0(\cdot \mid x)\right)\right] \\
&= \mathbb{E}_{x\sim\mathcal{D}}\left[\mathbb{E}_{y\sim\pi_\theta(\cdot|x)}\left[r_\phi(x,y)\right] - \beta \mathbb{E}_{y\sim\pi_\theta(\cdot|x)}\left[\log\frac{\pi_\theta(y \mid x)}{\pi_0(y \mid x)}\right]\right] \\
&= \mathbb{E}_{x\sim\mathcal{D}, y\sim\pi_\theta}\left[r_\phi(x,y) - \beta \log\frac{\pi_\theta(y \mid x)}{\pi_0(y \mid x)}\right]
\end{aligned}
$$

Robustness of reward optimization

Solutions:

1. Add KL penalty to the reward:
   (note that this is different from the KL penalty inside PPO)

$$
\begin{aligned}
J(\theta) &= \mathbb{E}_{x\sim\mathcal{D}}\left[\mathbb{E}_{y\sim\pi_\theta(\cdot|x)}\left[r_\phi(x,y)\right] - \beta \mathsf{KL}\left(\pi_\theta(\cdot \mid x) \| \pi_0(\cdot \mid x)\right)\right] \\
&= \mathbb{E}_{x\sim\mathcal{D}}\left[\mathbb{E}_{y\sim\pi_\theta(\cdot|x)}\left[r_\phi(x,y)\right] - \beta \mathbb{E}_{y\sim\pi_\theta(\cdot|x)}\left[\log \frac{\pi_\theta(y \mid x)}{\pi_0(y \mid x)}\right]\right] \\
&= \mathbb{E}_{x\sim\mathcal{D}, y\sim\pi_\theta}\left[r_\phi(x,y) - \beta \log \frac{\pi_\theta(y \mid x)}{\pi_0(y \mid x)}\right] \\
&= \mathbb{E}_{x\sim\mathcal{D}, y\sim\pi_\theta}\left[R_\phi(x,y)\right]
\end{aligned}
$$

Robustness of reward optimization

Solutions:

1. Add KL penalty to the reward:
   (note that this is different from the KL penalty inside PPO)

$$
\begin{aligned}
J(\theta) &= \mathbb{E}_{x\sim\mathcal{D}}\left[\mathbb{E}_{y\sim\pi_\theta(\cdot|x)}\left[r_\phi(x,y)\right] - \beta \mathrm{KL}\left(\pi_\theta(\cdot \mid x) \| \pi_0(\cdot \mid x)\right)\right] \\
&= \mathbb{E}_{x\sim\mathcal{D}}\left[\mathbb{E}_{y\sim\pi_\theta(\cdot|x)}\left[r_\phi(x,y)\right] - \beta \mathbb{E}_{y\sim\pi_\theta(\cdot|x)}\left[\log \frac{\pi_\theta(y \mid x)}{\pi_0(y \mid x)}\right]\right] \\
&= \mathbb{E}_{x\sim\mathcal{D}, y\sim\pi_\theta}\left[r_\phi(x,y) - \beta \log \frac{\pi_\theta(y \mid x)}{\pi_0(y \mid x)}\right] \\
&= \mathbb{E}_{x\sim\mathcal{D}, y\sim\pi_\theta}\left[R_\phi(x,y)\right]
\end{aligned}
$$

Rewarding trajectories that have high probability under π_0.

Robustness of reward optimization

Solutions:

1. Add KL penalty to the reward:
   (note that this is different from the KL penalty inside PPO)

$$
\begin{aligned}
J(\theta) &= \mathbb{E}_{x\sim\mathcal{D}}\left[\mathbb{E}_{y\sim\pi_\theta(\cdot|x)}\left[r_\phi(x,y)\right] - \beta \mathrm{KL}\left(\pi_\theta(\cdot \mid x) \| \pi_0(\cdot \mid x)\right)\right] \\
&= \mathbb{E}_{x\sim\mathcal{D}}\left[\mathbb{E}_{y\sim\pi_\theta(\cdot|x)}\left[r_\phi(x,y)\right] - \beta \mathbb{E}_{y\sim\pi_\theta(\cdot|x)}\left[\log \frac{\pi_\theta(y \mid x)}{\pi_0(y \mid x)}\right]\right] \\
&= \mathbb{E}_{x\sim\mathcal{D}, y\sim\pi_\theta}\left[r_\phi(x,y) - \beta \log \frac{\pi_\theta(y \mid x)}{\pi_0(y \mid x)}\right] \\
&= \mathbb{E}_{x\sim\mathcal{D}, y\sim\pi_\theta}\left[R_\phi(x,y)\right]
\end{aligned}
$$

   Rewarding trajectories that have high probability under π_0.

2. Early stop based on KL distance.

RLHF: Putting everything together

- Start with a pretrained language model
- **SFT model**: Finetune it on supervised data
- Collect human feedback on prompts and model outputs and train a **reward model**
- **RL model**: Optimize the reward on a set of prompts using PPO while monitoring KL distance between the RL model and the SFT model

Alternatives to RLHF

RLHF is a complicated process. What are simpler alternatives / baselines?

Alternatives to RLHF

RLHF is a complicated process. What are simpler alternatives / baselines?

- **SFT**. Instead of spending money on preference data, we can collect supervised data.
- **Best-of-n**. Use the reward model to rerank outputs.
- **Expert iteration**. Get best-of-n outputs, do SFT on it, and repeat.
- Other simpler RL algorithms.

Comparison of different approaches

[Dubois et al. 2023]

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
Best-of-n	45.0 ± 1.7	50.7 ± 1.8
Expert Iteration	41.9 ± 1.7	45.7 ± 1.7
SFT 52k (Alpaca 7B)	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

PPO is much better than SFT using roughly the same amount of data.

Comparison of different approaches

[Dubois et al. 2023]

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
Best-of-n	45.0 ± 1.7	50.7 ± 1.8
Expert Iteration	41.9 ± 1.7	45.7 ± 1.7
SFT 52k (Alpaca 7B)	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

Best-of-*n* has competitive performance. (What's a disadvantage of this method?)

Comparison of different approaches

[Dubois et al. 2023]

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
Best-of-n	45.0 ± 1.7	50.7 ± 1.8
Expert Iteration	41.9 ± 1.7	45.7 ± 1.7
SFT 52k (Alpaca 7B)	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

SFT performance saturate quickly with additional data.

Table of Contents

Motivation

- RLHF is difficult to get right (reward model, optimization stability, multiple moving pieces)
- Can we directly learn a policy from the preference data? (i.e. no reward model and no RL optimization)

Set up

- We have pairwise preference data (x, y_w, y_l) (assuming y_w is preferred over y_l)
- Can we learn a policy π_θ that maximizes $p(y_w \succ y_l)$?
- Recall: how do we model the probability?

$$p(y_w \succ y_l \mid x) = \frac{1}{1 + \exp(-(r(x, y_w) - r(x, y_l)))}$$

- Problem: the probability does not depend on the policy

Observation

- RL objective:

$$J(\theta) = \mathbb{E}_{x\sim\mathcal{D}, y\sim\pi_\theta}\left[r(x, y) - \beta \log \frac{\pi_\theta(y \mid x)}{\pi_0(y \mid x)}\right]$$

Observation

- RL objective:

$$J(\theta) = \mathbb{E}_{x \sim \mathcal{D}, y \sim \pi_\theta} \left[r(x, y) - \beta \log \frac{\pi_\theta(y \mid x)}{\pi_0(y \mid x)} \right]$$

- It's easy to show that the optimal policy under this objective is

$$\pi^*(y \mid x) = \frac{1}{Z(x)} \exp\left[\frac{1}{\beta} r(x, y) \right]$$

Observation

- RL objective:

$$J(\theta) = \mathbb{E}_{x \sim \mathcal{D}, y \sim \pi_\theta}\left[r(x, y) - \beta \log \frac{\pi_\theta(y \mid x)}{\pi_0(y \mid x)}\right]$$

- It's easy to show that the optimal policy under this objective is

$$\pi^*(y \mid x) = \frac{1}{Z(x)} \exp\left[\frac{1}{\beta} r(x, y)\right]$$

  - Exercise: show that the solution is the same as min KL $(\pi_\theta \| \pi^*)$

Observation

- RL objective:

$$J(\theta) = \mathbb{E}_{x \sim \mathcal{D}, y \sim \pi_\theta} \left[r(x, y) - \beta \log \frac{\pi_\theta(y \mid x)}{\pi_0(y \mid x)} \right]$$

- It's easy to show that the optimal policy under this objective is

$$\pi^*(y \mid x) = \frac{1}{Z(x)} \exp \left[\frac{1}{\beta} r(x, y) \right]$$

  - Exercise: show that the solution is the same as $\min \mathrm{KL}\left(\pi_\theta \| \pi^*\right)$
  - Why don't we directly use this optimal policy?

Observation

- RL objective:

$$J(\theta) = \mathbb{E}_{x \sim \mathcal{D}, y \sim \pi_\theta} \left[r(x, y) - \beta \log \frac{\pi_\theta(y \mid x)}{\pi_0(y \mid x)} \right]$$

- It's easy to show that the optimal policy under this objective is

$$\pi^*(y \mid x) = \frac{1}{Z(x)} \exp\left[\frac{1}{\beta} r(x, y) \right]$$

  - Exercise: show that the solution is the same as min KL $(\pi_\theta \| \pi^*)$
  - Why don't we directly use this optimal policy?
- This allows us to relate the reward and the policy

Observation

- RL objective:

$$J(\theta)= \mathbb{E}_{x\sim\mathcal{D},y\sim\pi_\theta}\left[r(x,y) - \beta \log \frac{\pi_\theta(y \mid x)}{\pi_0(y \mid x)}\right]$$

- It's easy to show that the optimal policy under this objective is

$$\pi^*(y \mid x) = \frac{1}{Z(x)} \exp\left[\frac{1}{\beta} r(x,y)\right]$$

  - Exercise: show that the solution is the same as min KL $(\pi_\theta \| \pi^*)$
  - Why don't we directly use this optimal policy?

- This allows us to relate the reward and the policy

- Therefore we can represent the reward using the policy in the objective

$$r^*(x,y) = \beta \log \frac{\pi^*(y \mid x)}{\pi_0(y \mid x)} + \beta \log Z(x)$$

New objective

- MLE objective on the preference dataset:

$$\min -\mathbb{E}_{(x,y_w,y_l)\sim\mathcal{D}} \log p_\theta(y_w \succ y_l \mid x) = -\mathbb{E}_{(x,y_w,y_l)\sim\mathcal{D}} \left[\frac{1}{1+\exp(-(r_\theta(x,y_w) - r_\theta(x,y_l)))} \right]$$

New objective

- MLE objective on the preference dataset:

$$\min -\mathbb{E}_{(x,y_w,y_l)\sim\mathcal{D}} \log p_\theta(y_w \succ y_l \mid x) = -\mathbb{E}_{(x,y_w,y_l)\sim\mathcal{D}} \left[\frac{1}{1+\exp(-(r_\theta(x,y_w) - r_\theta(x,y_l)))} \right]$$

- Representing $r_\theta(x, y)$ using $\pi_\theta(y \mid x)$

$$r_\theta(x,y) = \beta \log \frac{\pi_\theta(y \mid x)}{\pi_0(y \mid x)} + \beta \log Z(x)$$

Note that the objective only depends on the difference between the two rewards

New objective

- MLE objective on the preference dataset:

$$\min -\mathbb{E}_{(x,y_w,y_l)\sim\mathcal{D}} \log p_\theta(y_w \succ y_l \mid x) = -\mathbb{E}_{(x,y_w,y_l)\sim\mathcal{D}} \left[\frac{1}{1+\exp(-(r_\theta(x,y_w) - r_\theta(x,y_l)))} \right]$$

- Representing $r_\theta(x,y)$ using $\pi_\theta(y \mid x)$

$$r_\theta(x,y) = \beta \log \frac{\pi_\theta(y \mid x)}{\pi_0(y \mid x)} + \beta \log Z(x)$$

  Note that the objective only depends on the difference between the two rewards

- we get the DPO objective (σ is the logistic function)

$$\min -\mathbb{E}_{(x,y_w,y_l)\sim\mathcal{D}} \log \sigma \left(\beta \log \frac{\pi_\theta(y_w \mid x)}{\pi_0(y_w \mid x)} - \beta \log \frac{\pi_\theta(y_l \mid x)}{\pi_0(y_l \mid x)} \right)$$

What does DPO do?

$$\nabla_\theta \mathcal{L}_{\text{DPO}}(\theta) = -\,\beta\,\mathbb{E}_{(x,y_w,y_l)\sim\mathcal{D}}\left[\hat{p}_\theta(y_l \succ y_w)\left(\nabla_\theta \log \pi(y_w \mid x) - \nabla_\theta \log \pi(y_l \mid x)\right)\right],$$

where

$$\hat{p}_\theta(y_l \succ y_w) = \sigma\left(\beta \log \frac{\pi_\theta(y_l \mid x)}{\pi_0(y_l \mid x)} - \beta \log \frac{\pi_\theta(y_w \mid x)}{\pi_0(y_w \mid x)}\right)$$

- Increases the likelihood of the preferred response and decreases the likelihood of dispreferred response
- Large weight on the update if prediction is wrong

Summary

- RL had limited improvement over supervised learning in NLG on small models.
- Scaling helps boost performance of RL: large base model + large reward model
- But RL is still a complicated process in practice, and there are research towards simplifying the process (e.g., DPO).
- Key challenge:
  - Reward hacking / over-optimization
  - Unreliable human annotation