1.Alignment:

Large models learn a lot of knowledge and skills in pre-training, but not all knowledge and skills are what we want. For example, there is a common misconception that more than 80% of people have such a misconception, so this misconception will often appear in the pre-training data. Although there are also correct perceptions of this knowledge in the data set, the proportion is relatively low. If the model is allowed to directly use the knowledge learned in pre-training to answer, then the model may give wrong knowledge. This is not what we want. Therefore, some methods are needed to align the results given by the model with human preferences, such as the most basic preferences and correctness. Selecting the response and action we need from the very broad knowledge and skills of the model is the key to building a safe, efficient and controllable AI system. SFT is the most direct preference learning method, while RLHF/RLAIF is a preference alignment solution with a higher upper limit. However, RLHF is more complex, unstable in training, and expensive. The optimization goal of DPO is the same as RLHF, but it is simpler to implement.

2.RLHF:

Three steps in RLHF:

1.SFT phase

Based on pre-trained models, train on high-quality downstream task data and obtain π^{SFT}.

2.Reward Modeling Phase

First, given a prompt x, generate two answers $(y_1, y_2) \sim \pi^{\text{SFT}}(y|x)$, and compare y_1, y_2 through manual annotation to obtain the preference result (preference) $y_w \succ y_l \mid x$, where w and l represent win and lose.

Assume that among these preference results, there is a latent reward model $r^*(y, x)$ that we cannot directly access, and score each pair (x, y). This $r^*(y, x)$ is the fitting target of the reward model in RLHF.

Based on $r^*(y, x)$, there are many ways to model preference, and Bradley-Terry model is a common choice.

Distribution p^* of human preferences based on the Bradley-Terry model can be written as:

$$p^*(y_1 \succ y_2 \mid x) = \frac{\exp(r^*(x, y_1))}{\exp(r^*(x, y_1)) + \exp(r^*(x, y_2))}$$

Assume that we sample a static preference comparison dataset $D = \{x^{(i)}, y_w^{(i)}, y_l^{(i)}\}_{i=1}^N$from p^*, then we can use the reward model $r_\phi(x, y)$ initialized based on π^{SFT} to fit $r^*(y, x)$ by maximum likelihood. Formulating this problem as a binary classification problem, we get

the negative log-likelihood loss:

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

In order to ensure that the reward function has a low variance, the reward is generally normalized so that for all x, $\mathbb{E}_{x,y\sim\mathcal{D}} \left[r_\phi(x, y)\right] = 0$.

3.RL Fine-Tuning Phase

In the reinforcement learning phase, we use the reward obtained in the previous step to provide feedback to the target model and optimize the following goals

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

The first term in the above formula is the reward score given by the reward model to the answer given by the target model (i.e., the actor model in RLHF). The higher this term is, the better.

The second term is the KL divergence between the target model and the reference model, which is used to limit the trained target model from deviating too much from the reference model (i.e. π^{SFT}). This ensures that the reward model can work in a fully trained range, while preventing the target model from mode-collapse and loss of response diversity due to excessive optimization towards high reward scores. β is used to control the proportion of this restriction term.

Since language generation is discrete, the above optimization goal is not differentiable and needs to be optimized through RL.

The standard RL constructs the reward function as:

$$r(x, y) = r_\phi(x, y) - \beta(\log \pi_\theta(y \mid x) - \log \pi_{\mathrm{ref}}(y \mid x))$$

and optimize it through PPO.

3.Direct Performance Optimization

The goal of DPO is to derive a simple method that directly uses preferences for policy optimization without having to train a reward model.

3.1Derivation of the Goal of Direct Performance Optimization

First, the optimization objective of DPO is the same as that of RL: for any reward function $r(x, y)$, the reference model π_{ref}.

$$\max_{\pi} \mathbb{E}_{x\sim\mathcal{D},y\sim\pi} \left[r(x, y)\right] - \beta\mathbb{D}_{\mathrm{KL}} \left[\pi(y|x)||\pi_{\mathrm{ref}}(y|x)\right]$$

According to the definition of KL divergence, expand the second term in the above formula

$$\beta\mathbb{D}_{\mathrm{KL}}(\pi\|\pi_{\mathrm{ref}}) = \beta\sum_y \pi(y|x)\log\frac{\pi(y|x)}{\pi_{\mathrm{ref}}(y|x)}$$

The sum of conditional probabilities here is actually the expected value, so we have:

$$\max_\pi \mathbb{E}_{x\sim\mathcal{D},y\sim\pi}\left[r(x,y)\right] - \beta\mathbb{D}_{\mathbf{KL}}\left[\pi(y|x)||\pi_{\mathrm{ref}}(y|x)\right] = \max_\pi \mathbb{E}_{x\sim\mathcal{D}}\mathbb{E}_{y\sim\pi(y|x)}\left[r(x,y) - \beta\log\frac{\pi(y|x)}{\pi_{\mathrm{ref}}(y|x)}\right]$$

Then we can convert the maximization problem into a minimization problem

$$\begin{aligned}
&\max_\pi \mathbb{E}_{x\sim\mathcal{D}}\mathbb{E}_{y\sim\pi(y|x)}\left[r(x,y) - \beta\log\frac{\pi(y|x)}{\pi_{\mathrm{ref}}(y|x)}\right] \\
&= \min_\pi \mathbb{E}_{x\sim\mathcal{D}}\mathbb{E}_{y\sim\pi(y|x)}\left[\log\frac{\pi(y|x)}{\pi_{\mathrm{ref}}(y|x)} - \frac{1}{\beta}r(x,y)\right] \\
&= \min_\pi \mathbb{E}_{x\sim\mathcal{D}}\mathbb{E}_{y\sim\pi(y|x)}\left[\log\frac{\pi(y|x)}{\pi_{\mathrm{ref}}(y|x)\exp\left(\frac{1}{\beta}r(x,y)\right)}\right]
\end{aligned}$$

Here we use the partition function to normalize the denominator. Let:

$$Z(x) = \sum_y \pi_{\mathrm{ref}}(y|x)\exp\left(\frac{1}{\beta}r(x,y)\right)$$

Then we get a new effective probability distribution:

$$\pi^*(y|x) = \frac{1}{Z(x)}\pi_{\mathrm{ref}}(y|x)\exp\left(\frac{1}{\beta}r(x,y)\right)$$

Then we have:

$$\begin{aligned}
&\min_\pi \mathbb{E}_{x\sim\mathcal{D}}\mathbb{E}_{y\sim\pi(y|x)}\left[\log\frac{\pi(y|x)}{\pi_{\mathrm{ref}}(y|x)\exp\left(\frac{1}{\beta}r(x,y)\right)}\right] \\
&= \min_\pi \mathbb{E}_{x\sim\mathcal{D}}\mathbb{E}_{y\sim\pi(y|x)}\left[\log\frac{\pi(y|x)}{\frac{1}{Z(x)}\pi_{\mathrm{ref}}(y|x)\exp\left(\frac{1}{\beta}r(x,y)\right)} - \log Z(x)\right] \\
&= \min_\pi \mathbb{E}_{x\sim\mathcal{D}}\mathbb{E}_{y\sim\pi(y|x)}\left[\log\frac{\pi(y|x)}{\pi^*(y|x)} - \log Z(x)\right]
\end{aligned}$$

Because $Z(x)$ is not a function of y. We can take it out.

$$\begin{aligned}
&\min_{\pi} \mathbb{E}_{x\sim\mathcal{D}}\mathbb{E}_{y\sim\pi(y|x)}\left[\log\frac{\pi(y|x)}{\pi^*(y|x)} - \log Z(x)\right] \\
&= \min_{\pi} \mathbb{E}_{x\sim\mathcal{D}}\left[\mathbb{E}_{y\sim\pi(y|x)}\left[\log\frac{\pi(y|x)}{\pi^*(y|x)}\right] - \log Z(x)\right] \\
&= \min_{\pi} \mathbb{E}_{x\sim\mathcal{D}}\left[\mathbb{D}_{\mathrm{KL}}(\pi(y|x) \mid\mid \pi^*(y|x)) - \log Z(x)\right]
\end{aligned}$$

$Z(x)$ is not related to π, so to minimize this formula, we only need to minimize the first term, KL divergence. And if and only if the two distributions are exactly the same, KL divergence reaches the minimum value of 0, so we have.

$$\pi(y|x) = \pi^*(y|x) = \frac{1}{Z(x)}\pi_{\mathrm{ref}}(y|x)\exp\left(\frac{1}{\beta}r(x,y)\right)$$

Continue to make some changes to this formula

$$\pi_r(y|x) = \frac{1}{Z(x)}\pi_{\mathrm{ref}}(y|x)\exp\left(\frac{1}{\beta}r(x,y)\right)$$

$$\log Z(x) + \log \pi_r(y|x) = \log \pi_{\mathrm{ref}}(y|x) + \frac{1}{\beta}r(x,y)$$

$$r(x,y) = \beta\log\frac{\pi_r(y \mid x)}{\pi_{\mathrm{ref}}(y \mid x)} + \beta\log Z(x)$$

And we substitute the formula into the Bradley-Terry model, we can obtain:

$$p^*(y_1 \succ y_2 \mid x) = \frac{1}{1+\exp\left(\beta\log\frac{\pi^*(y_2|x)}{\pi_{\mathrm{ref}}(y_2|x)} - \beta\log\frac{\pi^*(y_1|x)}{\pi_{\mathrm{ref}}(y_1|x)}\right)}$$

Without going through the reward model, we can use MLE to optimize the target model directly on this probability model

$$\mathcal{L}_{\mathrm{DPO}}(\pi_\theta;\pi_{\mathrm{ref}}) = -\mathbb{E}_{(x,y_w,y_l)\sim\mathcal{D}}\left[\log\sigma\left(\beta\log\frac{\pi_\theta(y_w \mid x)}{\pi_{\mathrm{ref}}(y_w \mid x)} - \beta\log\frac{\pi_\theta(y_l \mid x)}{\pi_{\mathrm{ref}}(y_l \mid x)}\right)\right]$$

3.2Your Language Model Is Secretly a Reward Model

Under the Plackett-Luce preference framework, and in particular the BradleyTerry framework, two reward functions from the same equivalence class induce the same preference distribution.

If two reward function $r(x, y)$ and $r^{'}(x, y)$ can be written as:

$$r'(x, y) = r(x, y) + f(x)$$

This means that the two reward functions are from the same equivalence class.

For prompt x and answer $y_1, \ldots, y_K$, and the corresponding ranking τ, the proof under the Plackett-Luce framework (Bradley–Terry is also a special case) is as follows:

$$\begin{aligned}
p_{r'}(\tau|y_1, \ldots, y_K, x) &= \prod_{k=1}^{K} \frac{\exp(r'(x, y_{\tau(k)}))}{\sum_{j=k}^{K} \exp(r'(x, y_{\tau(j)}))} \\
&= \prod_{k=1}^{K} \frac{\exp(r(x, y_{\tau(k)}) + f(x))}{\sum_{j=k}^{K} \exp(r(x, y_{\tau(j)}) + f(x))} \\
&= \prod_{k=1}^{K} \frac{\exp(f(x)) \exp(r(x, y_{\tau(k)}))}{\exp(f(x)) \sum_{j=k}^{K} \exp(r(x, y_{\tau(j)}))} \\
&= \prod_{k=1}^{K} \frac{\exp(r(x, y_{\tau(k)}))}{\sum_{j=k}^{K} \exp(r(x, y_{\tau(j)}))} \\
&= p_r(\tau|y_1, \ldots, y_K, x)
\end{aligned}$$

Based on this, we can ignore the above $\beta \log Z(x)$ term, that is, the following two reward functions have the same preference distribution:

$$r(x, y) = \beta \log \frac{\pi_r(y \mid x)}{\pi_{\text{ref}}(y \mid x)} + \beta \log Z(x)$$

$$\hat{r}_\theta(x, y) = \beta \log \frac{\pi_\theta(y|x)}{\pi_{\text{ref}}(y|x)}$$

Furthermore, two reward functions from the same equivalence class will lead to the same optimal policy for the same RL problem.

Prove: two reward function from same equivalence can lead to same optimal policy. Assume $r'(x, y) = r(x, y) + f(x)$, π_r and $\pi_r^{'}$ are their optimal policy, respectively. We have

that:

$$\begin{aligned}
\pi_{r'}(y|x) &= \frac{1}{\sum_y \pi_{\text{ref}}(y|x) \exp\left(\frac{1}{\beta} r'(x,y)\right)} \pi_{\text{ref}}(y|x) \exp\left(\frac{1}{\beta} r'(x,y)\right) \\
&= \frac{1}{\sum_y \pi_{\text{ref}}(y|x) \exp\left(\frac{1}{\beta}(r(x,y) + f(x))\right)} \pi_{\text{ref}}(y|x) \exp\left(\frac{1}{\beta}(r(x,y) + f(x))\right) \\
&= \frac{1}{\exp\left(\frac{1}{\beta} f(x)\right) \sum_y \pi_{\text{ref}}(y|x) \exp\left(\frac{1}{\beta} r(x,y)\right)} \pi_{\text{ref}}(y|x) \exp\left(\frac{1}{\beta} r(x,y)\right) \exp\left(\frac{1}{\beta} f(x)\right) \\
&= \frac{1}{\sum_y \pi_{\text{ref}}(y|x) \exp\left(\frac{1}{\beta} r(x,y)\right)} \pi_{\text{ref}}(y|x) \exp\left(\frac{1}{\beta} r(x,y)\right) \\
&= \pi_r(y|x),
\end{aligned}$$

Theorem 1 Restated

Assume, we have a reference model, such that $\pi_{ref}(y|x) > 0$ for all pairs of prompts x and answers y and a parameter $\beta > 0$. All reward equivalence classes, as defined in Section 5 can be represented with the reparameterization $r(x, y) = \beta \log \frac{\pi(y|x)}{\pi_{ref}(y|x)}$ for some model $\pi(y|x)$.

Proposition 1. Assume, we have a reference model, such that $\pi_{ref}(y|x) > 0$ for all pairs of prompts x and answers y and a parameter $\beta > 0$. Then every equivalence class of reward functions, as defined in Section 5, has a unique reward function $r(x, y)$, which can be reparameterized as $r(x, y) = \beta \log \frac{\pi(y|x)}{\pi_{ref}(y|x)}$ for some model $\pi(y|x)$.