北京大學
PEKING UNIVERSITY

Fine-Tuning Language Models with Reward Learning on Policy

arXiv https://arxiv.org/pdf/2403.19279.pdf

Jiayu Yao

Overview

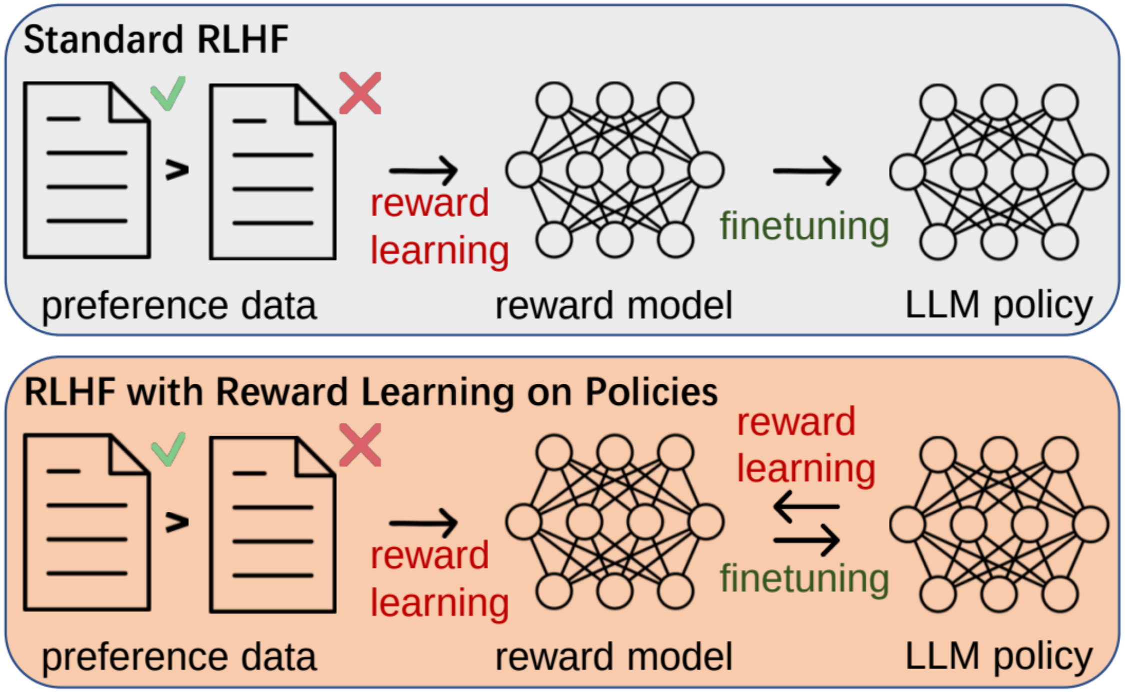

Figure 1: Comparison of standard RLHF (top) and RLHF with reward learning on policies (bottom). Different from (top), which performs reward learning and policy optimization serially, we iteratively train one of the two models with the help of the other.

RLHF

- Human Preference Collecting
- Reward learning

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

- RL policy optimization

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

Reward Learning on Policy

$$P = \left\{ (x, \boldsymbol{y}) \,|\, x \in \mathscr{U}, \boldsymbol{y} \sim \pi_\theta(y \,|\, x) \right\}$$

$$v_i = (x, \boldsymbol{y}) \,|\, \boldsymbol{y} \sim \pi_\theta$$

Unsupervised Multi-view learning

$$\mathcal{L}_M = \mathbb{E}_{(x, \boldsymbol{y}) \sim \mathcal{P}} [-\mathbb{I}(z_1; z_2) + \mathbb{D}_{\mathrm{SKL}}(p_\psi(z|v_1) || p_\psi(z|v_2))],$$

Synthetic Preference Generation

$$\hat{D} = \left\{ (x, y_w, y_l) \,|\, (x, \boldsymbol{y}) \in P, \frac{|\hat{\boldsymbol{g}}|}{|\boldsymbol{y}|} > \gamma, y_w \sim \hat{\boldsymbol{g}}, y_l \sim \boldsymbol{y}/\hat{\boldsymbol{g}} \right\}$$

$$\mathcal{L}_{\hat{R}} = -\mathbb{E}_{(x, y_w, y_l) \sim \mathcal{D} \cup \hat{\mathcal{D}}} [\log \sigma(\hat{r}_\phi(x, y_w) - \hat{r}_\phi(x, y_l))] \\ + \lambda \mathcal{L}_M,$$

Policy Retraining

Algorithm 1: RLP: RLHF with Reward Learning on Policy

Input: SFT model π^{SFT}, unlabeled data $\mathcal{U}$.

Output: A language model policy $\hat{\pi}_\theta$.

1. Collect a human preference dataset $\mathcal{D}$.
2. Train a reward model r_ϕ using $\mathcal{D}$.
3. Fine-tune a language model π_θ from π^{SFT} using $\mathcal{U}$ and r_ϕ.
4. Retrain a reward model $\hat{r}_\phi$ using $\mathcal{L}_{\hat{R}}$ (Eq. 1).
5. Fine-tune $\hat{\pi}_\theta$ from π^{SFT} using $\mathcal{U}$ and $\hat{r}_\phi$.

Proof

Method	AlpacaFarm Simulated Win-Rate	AlpacaFarm Human Win-Rate	LLMBar Simulated Win-Rate	Vicuna Simulated Win-Rate
GPT-4	79.0	69.8	74.0	85.0
ChatGPT	61.4	52.9	59.0	63.7
PPO	46.8	55.1	47.5	57.5
Best-of-n	45.0	50.7	43.4	52.5
SFT	36.7	44.3	42.4	50.0
LLaMA-7B	11.3	6.5	12.5	12.8
RLP-UML (ours)	49.1	56.5	48.5	61.3
RLP-SPG (ours)	**50.2**	**57.4**	**50.5**	**62.5**

Table 2: The win-rate (%) performance of RLP and baselines. Win-rates are computed against reference model `text-davinci-003`. Baseline results in AlpacaFarm come from Dubois et al. (2024). **Bold** numbers are superior results among the implemented LLMs. We omitted LLMBar and Vicuna for human evaluation because the simulated method rankings consistently correlate with the human method rankings in AlpacaFarm.

Thanks