Starling-7B: Increasing LLM Helpfulness & Harmlessness with RLAIF

Starling-7B: Increasing LLM Helpfulness & Harmlessness with RLAIF

Author: Banghua Zhu * , Evan Frick * , Tianhao Wu * , Hanlin Zhu and Jiantao Jiao

Starling-LM-7B (generated by DALL·E 3)

We introduce Starling-7B, an open large language model (LLM) trained by Reinforcement Learning from AI Feedback (RLAIF). The model harnesses the power of our new GPT-4 labeled ranking dataset, Nectar, and our new reward training and policy tuning pipeline. Starling-7B-alpha scores 8.09 in MT Bench with GPT-4 as a judge, outperforming every model to date on MT-Bench except for OpenAI's GPT-4 and GPT-4 Turbo. We release the ranking dataset Nectar, the reward model Starling-RM-7B-alpha and the language model Starling-LM-7B-alpha on HuggingFace, and an online demo in LMSYS Chatbot Arena. Stay tuned for our forthcoming code and paper, which will provide more details on the whole process.

*Based on MT Bench evaluations, using GPT-4 scoring. Further human evaluation is needed.

Overview

Supervised fine-tuning (SFT) has demonstrated remarkable effectiveness in developing chatbot systems from language models, particularly when leveraging high-quality data distilled from ChatGPT/GPT-4 (examples include Alpaca, Vicuna, OpenHermes 2.5, and Openchat 3.5). However, the extent to which Reinforcement Learning from Human Feedback (RLHF) or AI feedback (RLAIF) can enhance models when scaling high-quality preference data remains an open question. Earlier endeavors in the open-source community, such as Zephyra-7B, Neural-Chat-7B, and Tulu-2-DPO-70B, employed Direct Preference Optimization (DPO), but their performance in MT Bench (and some in Chatbot Arena), when compared to leading SFT models like OpenHermes 2.5 and Openchat 3.5, has not fully showcased RLHF's potential.

To facilitate more thorough research into RLHF, a high-quality ranking dataset specifically for chat is essential. We release Nectar, a GPT-4 labeled ranking dataset composed of 183K chat prompts. Each prompt includes 7 responses distilled from various models like GPT-4, GPT-3.5-instruct, GPT-3.5-turbo, Mistral-7B-Instruct, Llama2-7B, resulting in a total of 3.8M pairwise comparisons. Considerable effort was invested in mitigating positional bias when prompting GPT-4 for rankings, the details of which are elaborated in the dataset section below.

Moreover, there is a notable scarcity of open-source reward models. We address this gap by releasing our reward model Starling-RM-7B-alpha, trained with our K-wise loss on the Nectar dataset.

Lastly, we fine-tuned the Openchat 3.5 language model using the learned reward model. This resulted in an increase in the MT-Bench score from 7.81 to 8.09, and an improvement in the AlpacaEval score from 88.51% to 91.99%. Both metrics assess the chatbot's helpfulness.

We hope the open-sourced dataset, reward model and language model can help deepen the understanding of the RLHF mechanism and contribute to AI safety research. Our team is actively exploring various training methodologies for both the reward and language models, and will continue to update this blog with our findings and model releases.

Evaluation of the Model

Evaluating chatbots is never a simple task. We mainly evaluate the helpfulness of our models based on MT-Bench and AlpacaEval, which are GPT-4-based comparisons. We also test the basic capability of the model via MMLU. The results are listed below.

In line with findings in GPT-4 Technical Report, our observations post-RLHF reveal similar trends. We've observed improvements in the model's helpfulness and safety features; however, its basic capabilities in areas like knowledge-based QA, math, and coding have either remained static or experienced minor regression. We also detected a tendency for the model to respond with excessive caution to certain benign prompts after initial RLHF, while still showing vulnerabilities to jailbreaking attempts. This may require further fine-tuning with rule-based reward models with GPT-4 as classifiers, similar to what is done in the GPT-4 Technical Report. In the upcoming release of the paper, we will also benchmark the quality of the reward model, and the safety of the language model.

Model	Tuning Method	MT Bench	AlpacaEval	MMLU
GPT-4-Turbo	?	9.32	97.70	
GPT-4	SFT + PPO	8.99	95.28	86.4
Starling-7B	C-RLFT + APA	8.09	91.99	63.9
Claude-2	?	8.06	91.36	78.5
GPT-3.5-Turbo	?	7.94	89.37	70
Claude-1	?	7.9	88.39	77
Tulu-2-dpo-70b	SFT + DPO	7.89	95.1	
Openchat-3.5	C-RLFT	7.81	88.51	64.3
Zephyr-7B-beta	SFT + DPO	7.34	90.60	61.4
Llama-2-70b-chat-hf	SFT + PPO	6.86	92.66	63
Neural-chat-7b-v3-1	SFT + DPO	6.84	84.53	62.4
Tulu-2-dpo-7b	SFT + DPO	6.29	85.1	

The model is also currently included in LMSYS Chatbot Arena for both direct chat and anonymous comparisons for testing the human preferences. Please come and test it out!

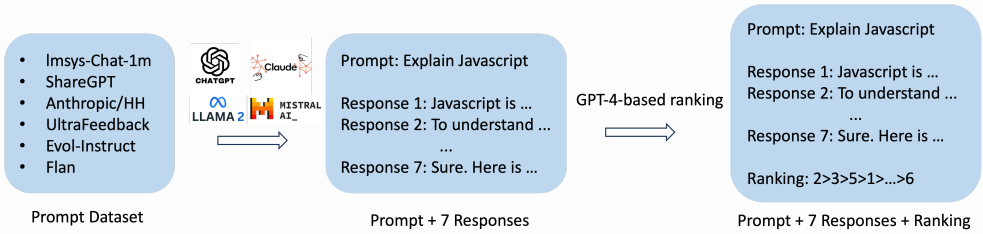
Comparisons of Benchmarks. In our evaluation of chat models, we've identified some limitations in using the Huggingface OpenLLM Leaderboard as a benchmark for chat model. Unlike Alpaca Eval and MT-Bench, the OpenLLM leaderboard doesn't support custom chat templates. This feature is crucial for nuanced model assessments, including Openchat 3.5, Llama 2 and other models which can be sensitive to chat template. Additionally, the OpenLLM leaderboard focuses on the basic capabilities of LLMs, while Alpaca Eval and MT Bench are designed for evaluating the chat assistants. Since RLHF doesn't inherently improve basic model capabilities, Alpaca Eval and MT-Bench are preferable for initial testing. Nevertheless, we believe the ultimate metric for model evaluation is human judgment, best exemplified by the LMSYS Chatbot Arena.

Goodhart's law for synthetic preference data. It's important to highlight that the model's preference ranking by GPT-4 does not necessarily correlate with human preference, a phenomenon that echoes the principles of Goodhart's Law. Essentially, a higher MT-Bench score, as endorsed by GPT-4, doesn't automatically imply greater human favorability, especially compared to models with lower scores. The core competencies of the model, encompassing basic knowledge, reasoning, coding, and mathematics,

remain unchanged. RLHF primarily enhances the style of the responses, in particular aspects of helpfulness and safety, as evidenced in its performance in MT-Bench and AlpacaEval. However, these results do hint at the potential of scaling online RL methods using extensive preference data. Our result shows that when the gold reward model is GPT-4's preferences, surpassing the performance of existing models is feasible with RLAIF. Therefore, adapting the preference data to include high-quality human responses could likely lead to improvements in aligning with human preferences.

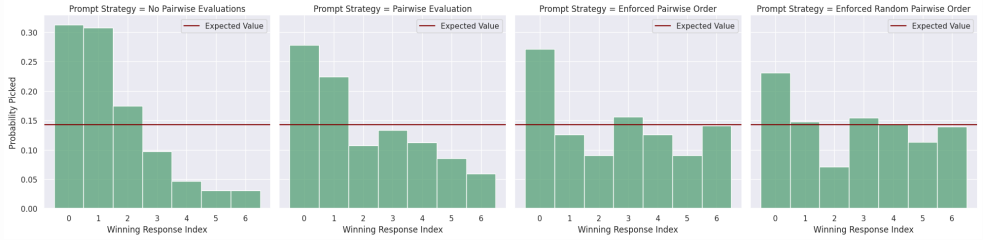
Dataset Overview

We present Nectar, the first high-quality 7-wise comparison dataset, generated through GPT-4-based ranking. For a high-quality RLHF dataset, one needs all the three components: diverse chat prompts, high-quality and diverse responses, along with accurate ranking labels. Our dataset's prompts are an amalgamation of diverse sources, including lmsys-chat-1M, ShareGPT, Antropic/hh-rlhf, UltraFeedback, Evol-Instruct, and Flan. Responses are primarily derived from a variety of models, namely GPT-4, GPT-3.5-turbo, GPT-3.5-turbo-instruct, LLama-2-7B-chat, and Mistral-7B-Instruct, alongside other existing datasets and models.



*Illustrating the creation process of Nectar, a 7-wise comparison dataset for RLAIF.

Overcoming Positional Bias. The most challenging aspect of creating Nectar was mitigating the positional bias inherent in GPT-4-based rankings. We extensively analyzed the likelihood of a response being selected as the top choice based on its position in the ranking prompt. Our initial findings, depicted in the first figure below, revealed a significant bias towards responses in the first and second positions when GPT-4 was simply asked to rank responses without additional reasoning.



*The positional bias of GPT-4-based ranking.

To address this, as shown in the second figure, we instructed GPT-4 to first conduct pairwise comparisons for all response pairs before compiling a 7-wise ranking. This approach moderately reduced the positional bias. We have also explored having GPT-4 score or judge each prompt individually before summarizing in a 7-wise ranking, but this method did not effectively diminish the bias.

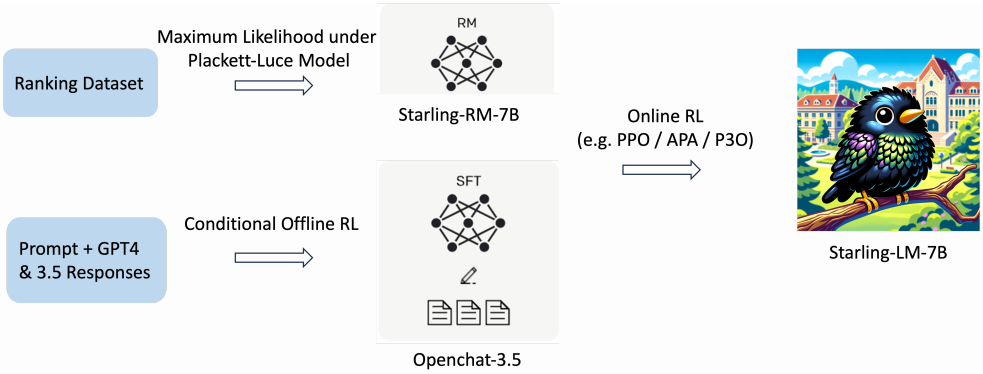
Further reduction of positional bias came with the introduction of a specific, and then a randomized pairwise evaluation order, as demonstrated in the third and fourth figures, respectively. This approach proved most effective in counteracting positional bias, leading to the final methodology employed in curating the Nectar dataset. Further details regarding dataset preparation and analysis will be elaborated in our upcoming paper.

We believe that Nectar will be a valuable resource for developers aiming to train more effective models using RLHF / RLAIF. It also offers high-quality responses for a diverse range of prompts, and can provide researchers with deeper insights into RLHF / RLAIF and the interplay between synthetic and human data.

RLHF / RLAIF

Reward Model Training

We train a reward model and conducting online RL based on the existing Nectar Dataset. Detailed below is our process, illustrated for clarity.



*Illustrating the RLHF / RLAIF process.

Our reward model is fine-tuned from Llama2-7B-Chat, and leverages the K-wise maximum likelihood estimator under the Plackett-Luce Model, as detailed in our prior paper. We discovered that for 7-wise comparisons, this new estimator yields a more effective reward model than the original loss, which converts comparisons into pairwise and minimizes cross-entropy loss.

Policy Finetuning

We selected Openchat 3.5 as the initial model for policy-finetuning, owing to its high MT Bench score (7.81). Our objective was to ascertain whether RLHF could enhance this score further. We experimented with an offline RL method Direct Preference Optimization (DPO), and three online RL methods: Advantage-induced Policy Alignment (APA), Proximal Policy Optimization (PPO), and Pairwise Proximal Policy Optimization (P3O).

DPO is simpler in implementation, which directly updates the language model on the pre-collected offline preference dataset. In contrast, online RL methods like PPO sample new responses using the current language model, score the new responses with the trained reward model, and update the language model with the reward information on the new responses. Despite the challenges in hyperparameter optimization for PPO, we found that, with optimal hyperparameter settings, the online RL methods yielded comparably strong results. We ultimately selected a checkpoint from an APA run. Our preliminary experiment on DPO showed no significant improvements over the initial model Openchat 3.5. This is likely due to that Openchat 3.5 has already done Conditioned RL Fine-Tuning (C-RLFT), a different format of offline preference-based training, and offline RL methods may not be as effective as online RL with a high-quality reward model. In the future, we envision a better language model fine-tuning procedure being using (conditional) offline RL including DPO or C-RLFT to leverage reward information to create a strong initial model, and further improve the helpfulness and harmlessness with reward training and online RL.

In our current implementation of online RL methods, we only unfreeze the last 4 layers of the model, aiming for faster training speed. The model is trained on 8 A100 GPUs with batch size 28 and 10k steps in total. In the future we plan to experiment with LoRA or full-parameter fine-tuning. This advancement could further enhance the overall quality of the model. More details about training and implementation will be soon released with the code and paper.

We observed that the quality of the preference dataset and reward model significantly influence the results, more so than the policy tuning method itself. We encourage the development of better reward learning methods, and invite researchers and developers to contribute to better open-source preference dataset, and utilize our dataset for training and testing. We believe it's likely that our dataset Nectar can bring higher gain with a larger reward model and language model, according to the scaling laws of the reward model.

Evaluation of RLHF

Evaluating RLHF algorithms presents unique challenges, particularly in discerning whether performance gains are due to imitation of the best demonstration policies in offline-RL-based methods or extrapolations of new reward signal in online-RL-based methods. We advocate for testing RLHF algorithms on our dataset, starting with models already proficient in learning from demonstrations, like Openchat 3.5. The ultimate benchmark should be the creation of models that surpass the initial model in both GPT-4 and human preferences.

However, training on GPT-4 preference data and evaluating against GPT-4-based scoring may invoke double layers of impact from Goodhart's laws. Over-optimization towards GPT-4 preferences could inadvertently harm actual human preferences. Similarly, the reward model, being a proxy for GPT-4 preference, might also misalign with GPT-4 preference itself when over-optimized. The challenge lies in

effectively utilizing synthetic preference data to mitigate these issues and evaluating models with minimal human intervention.

Limitations

Starling-7B, akin to other small-sized LLMs, has its limitations. It struggles with tasks involving reasoning or mathematics and may not always accurately self-identify or ensure the factual correctness of its outputs. Additionally, it's susceptible to jailbreaking prompts, as it wasn't explicitly trained for these scenarios. We also observe that in rare cases, the model may generate verbose or unnecessary content. We are committed to improving Starling-7B, exploring new reward training and policy training methods. We invite the community to collaborate with us in this endeavor to further improve the open dataset, reward models and language models with RLHF.

License

The dataset, model and online demo is a research preview intended for non-commercial use only, subject to the data distillation License of LLaMA, Terms of Use of the data generated by OpenAI, and Privacy Practices of ShareGPT. Please contact us if you find any potential violation.

Acknowledgment

We would like to thank Wei-Lin Chiang from Berkeley for detailed feedback of the blog and the projects. We would like to thank the LMSYS Organization for their support of lmsys-chat-1M dataset, evaluation and online demo. We would like to thank the open source community for their efforts in providing the datasets and base models we used to develope the project, including but not limited to Anthropic, Llama, Mistral, Hugging Face H4, LMSYS, OpenChat, OpenBMB, Flan and ShareGPT.

✉ **Correspondence to:** Banghua Zhu (banghua@berkeley.edu).

Citation

```
@misc{starling2023,
    title = {Starling-7B: Improving LLM Helpfulness & Harmlessness with RLAIF},
    author = {Zhu, Banghua and Frick, Evan and Wu, Tianhao and Zhu, Hanlin and Jiao, Jia
    month = {November},
    year = {2023}
}
```