Reinforcement Learning for Question Answering in Programming domain using Public Community Scoring as a Human Feedback

Extended Abstract

Alexey Gorbatovski
ITMO University
Saint Petersburg, Russia
gorbatoski@itmo.ru

Sergey Kovalchuk
Huawei
Saint Petersburg, Russia
sergey.kovalchuk@huawei.com

ABSTRACT

This study explores improving GPT Neo 125M in programming-focused Community Question Answering (CQA) using Reinforcement Learning from Human Feedback (RLHF) and Stack Overflow scores. We utilized two reward model training strategies with Proximal Policy Optimization (PPO), achieving enhancements comparable to GPT Neo's 2.7B model. The research introduces an auxiliary scoring mechanism, revealing the limitations of traditional linguistic metrics for programming responses. It highlights the need for domain-specific evaluation methods and the challenges in applying RLHF to programming CQA, contributing to the advancement of Large Language Models (LLMs) with human feedback.

KEYWORDS

Programming Question Answering; Reinforcement Learning from Human Feedback; Stack Overflow; Natural Language Processing; Reinforcement Learning; Proximal Policy Optimization

ACM Reference Format:

Alexey Gorbatovski and Sergey Kovalchuk. 2024. Reinforcement Learning for Question Answering in Programming domain using Public Community Scoring as a Human Feedback: Extended Abstract. In *Proc. of the 23rd International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2024), Auckland, New Zealand, May 6 – 10, 2024*, IFAAMAS, 3 pages.

1 INTRODUCTION

This paper addresses the challenges and potential of RLHF in refining the response generation of Large Language Models (LLMs) in programming-focused Community Question Answering (CQA). While RLHF has enhanced LLMs in general contexts [9], its effectiveness in complex domains like programming, involving tasks like conceptual understanding and code generation, is less studied [2].

A key issue is the inadequacy of conventional evaluation metrics like BertScore and Rouge in capturing the nuances of programming responses, necessitating more semantically-accurate metrics [8, 12]. Our research explores applying RLHF to GPT Neo 125M [4] in programming CQA, aiming to improve response quality and develop better evaluation methods. We demonstrate the effectiveness of RLHF in training smaller LLMs for programming tasks and reveal the gap between traditional metrics and RLHF reward models, underscoring the need for improved evaluation techniques.

2 BACKGROUND AND DATASET

Reinforcement Learning from Human Feedback. RLHF enhances LLMs by using human feedback as rewards, beginning with supervised fine-tuning (SFT) and involving a separate reward model for optimization [9, 11, 13]. While RLHF has been applied in various contexts, its implementation in programming CQA remains underexplored. The objective of RLHF with PPO [10] is formalized as: $\max_\theta \mathbb{E}_{x\sim D, y\sim\pi_\theta(y|x)} \left[r_\phi(x,y) - \beta D_{KL}[\pi_\theta(y|x) \parallel \pi_{\text{ref}}(y|x)]\right]$.

Here, θ are the model parameters, $\pi_\theta(y|x)$ the policy, $r_\phi(x,y)$ the reward model, and β a scaling factor for the KL divergence D_{KL} between the learned policy π_θ and a reference policy π_{ref}.

Dataset. We utilized a Stack Overflow (SO) dataset[1], specifically targeting 'python' tagged content. The dataset, comprising titles, questions, answers, and user scores, was used for both SFT and partial reward model training. Selection criteria involved: focusing on 'API Usage' questions as per Beyer et al. [3]; excluding entries with images, links, or code blocks; transforming HTML to plain text for NLP compatibility.

This resulted in 6,946 training and 1,000 post-December 14, 2021, validation entries. The dataset's constraints, while necessary, may limit the diversity and real-world applicability of the questions.

3 METHODOLOGY

Users' Feedback Processing. In adapting RLHF for programming Q&A, we processed SO user ratings into two distinct types of feedback for reward model training: (1) *Regression Dataset:* Consists of Q&A prompts with user ratings. Ratings were normalized per question, outliers clipped at 1.5IQR, and scaled. This dataset evaluates individual response quality; (2) *Contrastive Dataset:* Contains pairs of answers for each question, contrasting each with the highest-rated response. Ratings were logarithmically scaled [1], with accepted answers receiving an additional increment. Negative ratings were set to -1.

SFT Data Generation. We generated 6,872 synthetic answers for single-response questions using the SFT model. These were included in the regression dataset with a distribution $\mathcal{N}(-0.5, 0.1^2)$,

More details on the presented research can be found on arXiv [7].

Proc. of the 23rd International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2024), N. Alechina, V. Dignum, M. Dastani, J.S. Sichman (eds.), May 6 – 10, 2024, Auckland, New Zealand. © 2024 International Foundation for Autonomous Agents and Multiagent Systems (www.ifaamas.org).

[1] https://stackoverflow.com/

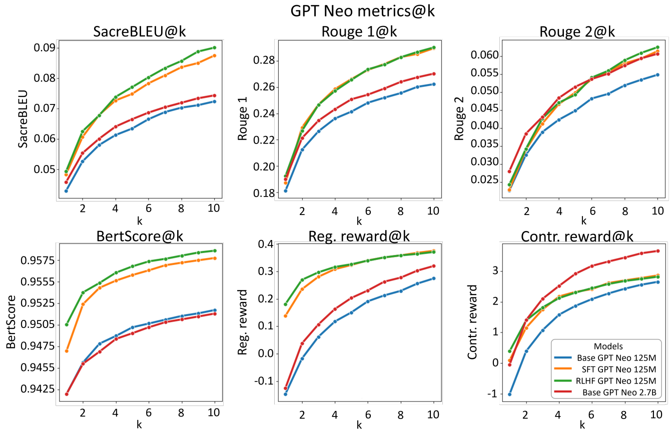

Figure 1: Graphs of dependencies of metric values on the number of k attempts to generate

indicating lower quality, and added to the contrastive dataset as negatively evaluated responses.

Reward Model Training. The GPT Neo 125M [5] models were trained using two approaches with a value head: regression and answer comparison. The regression method utilized Mean Squared Error for loss calculation, while answer comparison employed contrastive loss via the Bradley–Terry model [6]. Both methods were based on the SFT model.

QA Model Training. We utilized the GPT Neo model with 125 million parameters for the Q&A task. Initially, the model was fine-tuned through SFT, to adapt to the Q&A format. This was followed by RLHF, guided by reward models developed through regression and contrastive approaches. However, the model trained with the contrastive reward approach failed to converge. Consequently, the RLHF phase results predominantly reflect the outcomes using the regression-based reward model, which proved more effective in refining response generation capabilities.

4 RESULTS

Our experiments evaluated the efficacy of the RLHF approach in programming QA response generation, comparing Base GPT Neo 125M, SFT 125M, RLHF 125M, and Base GPT Neo 2.7B models. These models evaluated using several metrics, including SacreBLEU, Rouge 1, Rouge 2 and BertScore, as well as the scores obtained from the regression and contrastive reward models. Key findings include:

Comparison of Average Metrics: The RLHF 125M model demonstrated superior performance in SacreBLEU and BertScore metrics, indicative of enhanced response quality. In contrast, the Base 2.7B model excelled in Rouge scores. Statistically significant as per the KS-test, these metrics underscore the RLHF model's ability to generate more relevant words and provide better semantic context.

Metric@k Analysis: Analyzing the best scores from 10 generation attempts (metric@k), both SFT and RLHF models often outperformed Base 2.7B, especially in SacreBLEU, Rouge 1 and BertScore, suggesting better semantic alignment. The metric@k approach revealed that these models have a higher potential for generating quality responses compared to their larger counterpart (see Figure 1).

MRR Comparison: In evaluating the top 10 ranked responses for 100 questions using binary human assessment, Rouge 1 and 2 metrics showed high Mean Reciprocal Rank (MRR@10) scores, indicating their effectiveness in assessing response accuracy. The trained reward models outperformed SacreBLEU and BertScore, reflecting their refined evaluation capabilities (Table 1).

Table 1: Comparison of MRR@10 scores for different models and metrics. The values represent the MRR scores considering the top 10 ranked samples.

	Base 125M	**SFT 125M**	**RLHF 125M**
SacreBLEU	0.4107	0.3709	0.3262
Rouge 1	**0.4792**	**0.4532**	0.4091
Rouge 2	0.4011	0.4453	0.4220
BertScore	0.2913	0.3403	0.3300
Reg. Reward	0.4015	0.3867	**0.4296**
Contr. Reward	0.4302	0.3787	0.3527

Correlation Analysis: Spearman correlation analysis revealed a strong correlation between Rouge 1 and Rouge 2 rankings, while BertScore displayed minimal or negative correlation with other metrics, questioning its comparative reliability. The reward models trained via different methodologies showed minimal correlation amongst themselves, highlighting methodological impacts on evaluation outcomes.

These results collectively emphasize the RLHF training's role in enhancing response quality for programming QA, while also illuminating the complexities in metric evaluations.

5 CONCLUSION AND DISCUSSION

This study on RLHF's application in programming QA, focusing on GPT Neo 125M, demonstrates its efficacy over the SFT technique. Utilizing Stack Overflow data, we employed regression and contrastive scores with PPO, underscoring the value of real-world forum data in model training.

Key findings include the effectiveness of Rouge scores for response accuracy, contrasted with the ambiguities in BertScore and SacreBLEU metrics. These discrepancies, highlighted by near-zero Spearman correlations, suggest the inadequacy of traditional metrics for complex domains like programming. These domains are marked by intricate semantic relationships and a broad spectrum of valid answers.

The study also illustrates the potential of RLHF in efficiently training smaller models, with a caveat regarding the interpretability of certain evaluation metrics in complex semantic environments. Future work aims to explore these methodologies on larger models, assessing scalability and result consistency.

In summary, our research contributes initial insights into RLHF in programming CQA, emphasizing the importance of domain-specific evaluation methods. These findings are a step towards understanding and improving model performance in complex domains, highlighting the need for further research in this evolving field.

REFERENCES

[1] Amanda Askell, Yuntao Bai, Anna Chen, Dawn Drain, Deep Ganguli, Tom Henighan, Andy Jones, Nicholas Joseph, Ben Mann, Nova DasSarma, et al. 2021. A general language assistant as a laboratory for alignment. *arXiv preprint arXiv:2112.00861* (2021).

[2] Edward Beeching, Younes Belkada, Kashif Rasul, Lewis Tunstall, Leandro von Werra, Nazneen Rajani, and Nathan Lambert. 2023. StackLLaMA: An RL Fine-tuned LLaMA Model for Stack Exchange Question and Answering. https://doi.org/10.57967/hf/0513

[3] Stefanie Beyer, Christian Macho, Massimiliano Di Penta, and Martin Pinzger. 2020. What kind of questions do developers ask on Stack Overflow? A comparison of automated approaches to classify posts into question categories. *Empirical Software Engineering* 25 (2020), 2258–2301.

[4] Sid Black, Leo Gao, Phil Wang, Connor Leahy, and Stella Biderman. 2021. Gpt-neo: Large scale autoregressive language modeling with mesh-tensorflow. *If you use this software, please cite it using these metadata* 58 (2021).

[5] Sid Black, Leo Gao, Phil Wang, Connor Leahy, and Stella Biderman. 2021. *GPT-Neo: Large Scale Autoregressive Language Modeling with Mesh-Tensorflow.* https://doi.org/10.5281/zenodo.5297715 If you use this software, please cite it using these metadata.

[6] Ralph Allan Bradley and Milton E Terry. 1952. Rank analysis of incomplete block designs: I. The method of paired comparisons. *Biometrika* 39, 3/4 (1952), 324–345.

[7] Alexey Gorbatovski and Sergey Kovalchuk. 2024. Reinforcement learning for question answering in programming domain using public community scoring as a human feedback. arXiv:2401.10882 [cs.CL]

[8] Sergey V. Kovalchuk, Vadim Lomshakov, and Artem Aliev. 2022. Human perceiving behavior modeling in evaluation of code generation models. In *Proceedings of the 2nd Workshop on Natural Language Generation, Evaluation, and Metrics (GEM)*. Association for Computational Linguistics, Abu Dhabi, United Arab Emirates (Hybrid), 287–294. https://aclanthology.org/2022.gem-1.24

[9] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems* 35 (2022), 27730–27744.

[10] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347* (2017).

[11] Nisan Stiennon, Long Ouyang, Jeffrey Wu, Daniel Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul F Christiano. 2020. Learning to summarize with human feedback. *Advances in Neural Information Processing Systems* 33 (2020), 3008–3021.

[12] Qicai Wang, Peiyu Liu, Zhenfang Zhu, Hongxia Yin, Qiuyue Zhang, and Lindong Zhang. 2019. A text abstraction summary model based on BERT word embedding and reinforcement learning. *Applied Sciences* 9, 21 (2019), 4701.

[13] Daniel M Ziegler, Nisan Stiennon, Jeffrey Wu, Tom B Brown, Alec Radford, Dario Amodei, Paul Christiano, and Geoffrey Irving. 2019. Fine-tuning language models from human preferences. *arXiv preprint arXiv:1909.08593* (2019).