e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:07/Issue:01/January-2025 Impact Factor- 8.187 www.irjmets.com

OPTIMIZING MEDICAL DIALOGUE SYSTEMS WITH REINFORCEMENT LEARNING FROM HUMAN FEEDBACK ARCHITECTURES

Abdul Rehman[*1], Prof. Dr. Ali Okatan[*2]

[*1,2]Department Of Artificial Intelligence And Data Science, Istanbul Aydin University, Istanbul, Turkey.

DOI : https://www.doi.org/10.56726/IRJMETS65883

ABSTRACT

Reinforcement Learning from Human Feedback (RLHF) is a new way to make large language models (LLMs) better by aligning them with human preferences in medical chat systems and other specific fields. This research looks into two different RLHF setups. The first uses a Roberta reward model along with a GPT-2 generation model. The second uses a Llama3.1-8B reward model along with a Llama3.2-1B generation model. Roberta, which is meant to classify text, and Llama, a generative model famous for understanding context well, were tested to see how well they worked in the RLHF process. We took the medical conversation dataset and preprocessed it to create a pairwise comparison dataset with messages, accepted answers, and refused responses. The dataset was then cleaned up to make it easier to understand and all messages were made to fit within a 512-token limit. Both models were very good at telling the difference between answers that were accepted and those that were refused. They were added to an RLHF framework along with Proximal Policy Optimization (PPO). To make the generative models more in line with human preferences, they were also limited to a 512-token context as the messages within the dataset. To judge success, key indicators were looked at, such as reward scores and policy changes. Low-Rank Adaptation (LoRA) and Quantization were added to the Llama-based setup to make it more efficient. Roberta's strong classification made it easier for clear reward signals, but the limited contextual understanding made them less useful for steering GPT-2. On the other hand, the Llama-based process created results that fit together better and were more relevant to the situation. This study shows the trade-offs of using classification-based and context-aware reward models in RLHF systems. It stresses how important it is to use high-quality datasets and training methods that don't use too many resources to get the best results from LLMs.

Keywords: Reinforcement Learning, RLHF, GPT-2, Llama, Medical Dialogue Systems, Proximal Policy Optimizations.

I. INTRODUCTION

Large language models (LLMs) have significantly advanced natural language processing (NLP). These models are driving innovation in areas like healthcare, education, business, and law. LLMs are very smart, but they often can't handle domain-specific tasks and need a lot of thinking and understanding of the situation. Even when used on very large datasets, traditional supervised learning methods can produce results that are scientifically wrong, don't make sense in the given situation, or are morally questionable. These flaws are especially worrying in private areas where accuracy and trust are very important. Reinforcement Learning from Human Feedback (RLHF) gets around these problems by using human preference in the training process itself. RLHF is different from standard guided methods because it uses a reward model that understands how humans evaluate things, which lets LLMs put clarity, applicability, and accuracy at the top. RLHF lets generative models change their results to better fit the needs of a given task by using Proximal Policy Optimization (PPO) for repeated refinement. This method works really well in medical situations where being responsible, knowing the situation well, and making accurate diagnoses are very important. This study looks into how RLHF can be used in medical interaction systems using two different setups: a Roberta-based reward model leading a GPT-2 generating model and a Llama-based reward model paired with a smaller Llama generative model. To learn from human preferences, these reward models are fine-tuned by comparing sets of accepted and refused answers one at a time. The RLHF system uses these reward models to make sure that the generated results meet medical standards. The study aims to improve RLHF processes for medical conversation by looking at how well these two pipelines work. This will lead to better evaluations, better communication between patients, and solid insights that can be used. Although RLHF has significant potential for aligning LLMs with human preferences in medical dialogues, it is not without challenges. For training, the need for large amounts of

e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:07/Issue:01/January-2025 Impact Factor- 8.187 www.irjmets.com

computer resources is a huge problem. When RLHF is mixed with reward models and PPO, it needs a lot of GPU time, memory, and processing power, which makes the training process very resource-intensive. Also, because RLHF is an iterative process, optimization, and exploration have to be done many times. This makes training processes long, which takes a lot of time and costs a lot of money. Another problem is that there aren't many high-quality, domain-specific samples for medical conversations. Existing datasets are often small and lack the depth and accuracy needed for good training. This makes it harder for RLHF to get the best results in this area. Getting past these problems is important for RLHF to be used more often in medical NLP uses.

II. METHODOLOGY

This section gives an outline of the data set and study methods that were used. It starts by explaining the preparation and preliminary analysis that were done to make sure that the data for the experiments was of good quality. Key preparatory procedures, including as tokenization and the development of paired datasets optimized for the RLHF pipeline, are described. It goes into great depth about the RLHF designs Roberta-GPT-2 and Llama-Llama. Proximal Policy Optimization (PPO) is used by both designs to make sure that model results are in line with human preferences.

Dataset

The ReMeDi dataset addresses problems that come up when training machine learning models with medical conversations. It has 96,965 conversations between doctors and patients about evaluation, advice, and treatment in 40 different areas of medicine. These areas cover a huge range of medical topics, including 843 illnesses and 5,228 medical issues. To make the statistics better, short conversations with less than eight utterances and multimedia were filtered out. Sampling made sure that each disease was fairly represented, which made the collection more useful for training and testing. The layout of the dataset records real-life medical exchanges that make model training more reliable.

Data Preprocessing

During preprocessing, dialogues were organized into role-based conversations, and the speaker's part (as a doctor or a patient) and message text were extracted. These were put into a Python dictionary where the part of the speaker was the key and the message was the value. This step got rid of things that weren't important and focused on the important conversations. During the filtering process, system-generated or system-error messages were also deleted, such as automatic explanations. Afterward, the dialogues were presented as lists of objects, with each object representing a message. As shown in Figure 1, a histogram study showed that most conversations had less than 25 messages.

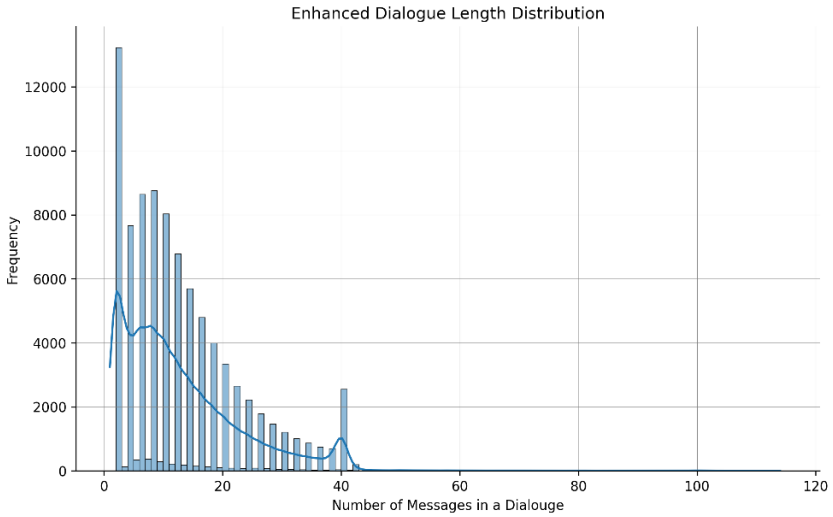

Figure 1: Distribution of Dialogue Lengths within the Dataset.

By combining repeated messages from the same person, the natural flow of the talk was improved. This made the dialogue structure more logical. In Figure 2, we can see the graph of the improved conversations. The conversations were also changed from lists to strings while keeping the format of "User Role followed by the text." Further filtration limited conversations to eight messages at most, which improved data quality and cut

e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:07/Issue:01/January-2025 Impact Factor- 8.187 www.irjmets.com

down on biases. After being cleaned up, the preprocessed data was shuffled up to make it more random and then saved for later use.

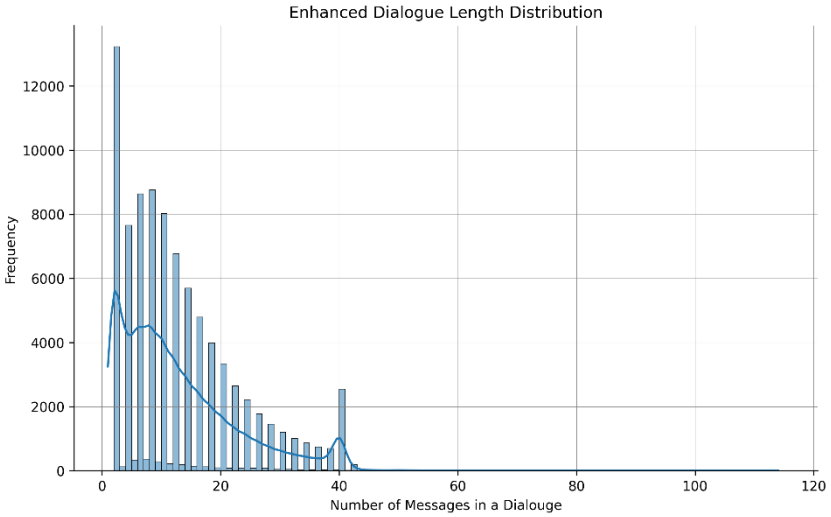

Figure 2: Distribution of Enhanced Dialogue Lengths within the Dataset.

Pairwise Dataset Preparation

In the RLHF process, the labels were set to "Assistant" for doctors and "User" for patients so that the reward model could be trained. For clarity, unfinished conversations were filtered out, like those that ended suddenly with questions ignored. A pandas DataFrame was set up with three columns: "rejected" (less accurate answers), "accepted" (confirmed doctor responses), and "prompt" (patient input). We used Llama-3 as our base model. It was used as a supervised fine-tuned model that gave us rejected answers for the "rejected" column. Finally, the contextual history from doctor-patient interactions was included in prompts to improve understanding and eliminate hallucinations throughout the answer generation process.

Dataset Tokenization

Tokenizing the text into vectors for language models was the last step. To make the training process efficient and use optimal computer resources, a context limit of 512 tokens was set. Text that went over this limit was cut off, and conversations with more than 500 words were filtered out. The finished dataset was cut down to about 10,000 high-quality data points, and it was optimized for training the reward model and fine-tuning the language model with RLHF.

Reward Models

Reward models are an important part of Reinforcement Learning from Human Feedback (RLHF). They help language models make decisions that are in line with human preferences. The data that these models are trained on has been labeled by humans so that they can rate and evaluate responses. This input changes the reinforcement learning process. Reward models make sure that AI systems generate output that meets standards for correctness, fluency, and usefulness by improving outputs over and over again. In RLHF, reward models grade language model results by giving them scores. They are a middle ground between guided fine-tuning, which uses clear labels, and reinforcement learning, which focuses on long-term goals. Reward models give a scalar reward to measure the quality of the generated output to a certain input. This feedback makes the language model's policy more in line with human preferences. Using pairwise data is a popular way to train the reward models. Annotators look at more than one response to the same input query and rank them, they decide which one is accepted and which one is rejected. With this method, the reward model can learn to tell the difference between different types of responses. The labeled data is then used to train the reward model to predict the preferred response from the given pair of data. Mathematically, this can be expressed as:

$$x \rightarrow y \quad \text{where} \quad y \in \{y_0, y_1\} \tag{1}$$

Given x as a query having a response y_i that is preferred by the human, we can write the loss function of reward model as:

$$\text{loss}(r_\theta) = -E_{(x,y_0,y_1,i)\sim D}\left[\log\left(\sigma\left(r_\theta(x,y_i) - r_\theta(x,y_{1-i})\right)\right)\right] \tag{2}$$

e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:07/Issue:01/January-2025 Impact Factor- 8.187 www.irjmets.com

Here in equation 2 the $r_\theta(x, y)$ is the scalar output (reward score) from the reward model for the human query x and response y with parameters θ. The D in the equation represents the entire dataset. At the end of the training process of the reward model the ouptut scores are normalized. Most reward models are based on pre-trained transformer architectures such as GPT and BERT and are fine-tuned to predict reward scores. The queries and responses are combined together into a single input sequence, then the combined input is passed through the transformer encoder to make contextualized embeddings. These embeddings are then passed through a neural network with a linear output layer to predict a reward score. This approach makes it possible for reward models to work with different language models and adapt to different scenarios.

Proximal Policy Optimization

Proximal Policy Optimization is a well-known reinforcement learning optimization method and it is good at finding the best balance between speed and stability. PPO is a simple optimization algorithm that ensures stable learning and computational efficiency, which makes it perfect for fine-tuning complex large language models. In order to maximize expected rewards, PPO, a policy gradient approach, modifies the probability distribution of actions. In order to maintain stability and avoid performance collapse, it imposes limits to stop overcorrection during updates. PPO also makes efficient use of data by changing the policy more than once even with the same batch, this approach reduces the need to collect an extensive amount of data. This makes PPO stable, good at using samples, and scalable for high-dimensional models like transformers. PPO is based on a surrogate objective function that strikes a balance between optimization and stability. The policy ratio is an important part of PPO because it compares the probabilities that the new policy gives to the old policy for the same actions. This can be expressed as:

$$r(\theta) = \frac{\pi_\theta(a|s)}{\pi_{\theta_k}(a|s)} \quad (3)$$

In the equation 3 above the $\pi_\theta(a|s)$ represents the updated policy probabilities and $\pi_{\theta_k}(a|s)$ represents the old policy probabilities of action a in state s . In the PPO objective we have a mechanism known as clipping mechanism that prevents large deviations in the policy ratio. PPO-clip updates the policy using the equation given below:

$$L(s, a, \theta_k, \theta) = \min\left(r(\theta)\, A^{\pi_{\theta_k}}(s, a),\ \text{clip}(r(\theta), 1 - \epsilon, 1 + \epsilon) A^{\pi_{\theta_k}}(s, a)\right) \quad (4)$$

There is a small hyperparameter ϵ within the equation, it controls how far the new policy can be different from the old one. The clipping method makes sure that updates stay stable and don't change too much. By keeping this balance between exploration and exploitation, PPO makes sure that language models in the RLHF framework can be fine-tuned in an effective and efficient way.

III. EXPERIMENT

The Llama-3 model was chosen as the base supervised fine-tuned model for the project because it is very good at natural language understanding. Llama-3 generated contextually accurate answers that make sense without any extra supervised fine-tuning because it has already been trained on general language data. In the next step, a pairwise collection of accepted and refused answers is used to build a reward model. This reward model rates the quality of the text based on medical logic and how clear it is. The RLHF pipeline includes the generative language model, the reward model, and Proximal Policy Optimization (PPO). It updates the parameters of the generative model to give results that are in line with human preferences.

Rlhf Pipeline With Roberta Reward Model And Gpt-2 Generation Model

A Roberta-based reward model and a GPT-2 generative model are both part of this process. The reward model is based on the Distilroberta Base design, which is best at classifying the difference between good (accepted) and bad (rejected) medical query answers. The input data is tokenized using a custom formatting function that is set up with truncation, padding, and a maximum token length of 512 to make sure it works with the model's context size. A batch size of 16 is used for supervised learning to train the reward model. Every 50 steps, there is logging and evaluation, and checkpoints are saved at each interval. The trained reward model gives answers as scalar numbers that show how well they match up with the input questions. These results tell GPT-2 how to change the parameters in the RLHF process. The GPT-2 model takes medical conversation data that has already been cleaned up and comes up with answers for each question. These answers are scored by the reward model,

e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:07/Issue:01/January-2025 Impact Factor- 8.187 www.irjmets.com

and PPO changes GPT-2's settings based on the reward scores. A learning rate of 1.41e-5 and a batch size of 16 are used in the training process to make sure that optimization is fair. To keep track of the training process, metrics like reward mean, policy loss, and value loss are monitored.

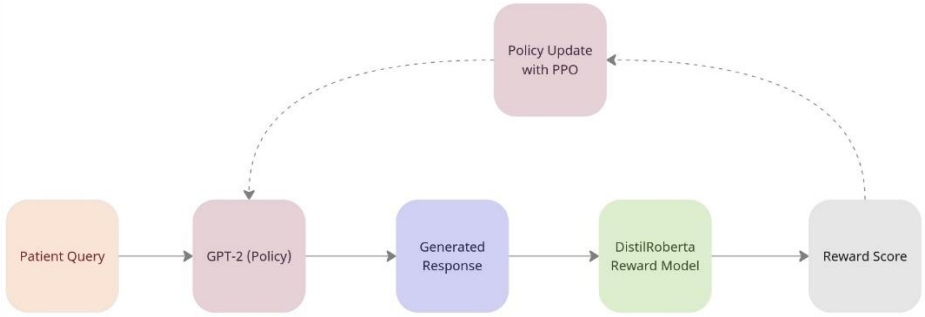

Figure 3: Architecture of Roberta Reward & GPT-2 Generative Model.

Rlhf Pipeline With Llama Reward And Generation Model

In this process, a Llama-3.1 reward model with 8 billion parameters and a Llama-3.2 generation model with 1 billion parameters were used. Parameter-efficient fine-tuning (PEFT) and 8-bit quantization are used to control the amount of computing that needs to be done. The reward model rates pairs of prompts and responses, which helps the generating model get better. LoRA (Low-Rank Adapters) tweaks a small group of the model's weights, which makes it use less memory and computing power. Tokenization choices, such as truncation and padding with a maximum length of 512 tokens, keep input consistent. A regression goal that gives out scalar values that show reward quality is used to build the reward model. Because the reward model is so big, the batch size is limited to two, and 8 gradient accumulation steps make sure that updates are stable. The Llama-3.2-1B generative model is optimized by PPO. The model comes up with an answer to each question that is then judged by the Llama-3.1-8B reward model. Based on the reward scores, PPO changes the settings of the generative model. It uses a learning rate of 1.41e-4 and a batch size of 16, which makes training effective.

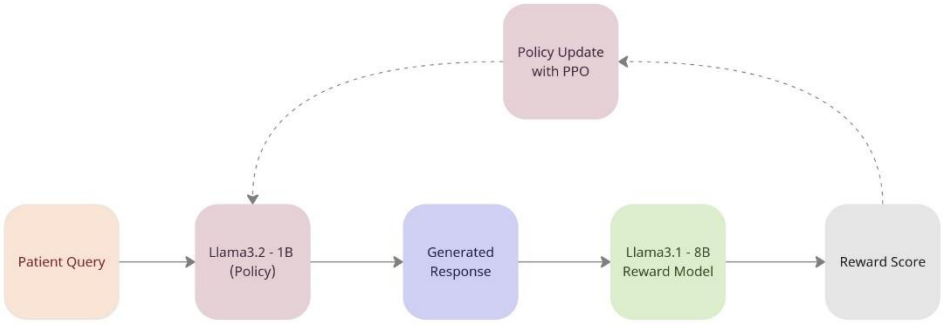

Figure 4: Architecture of Llama3.1-8B Reward & Llama3.2-1B Generative Model.

Evaluation Metrics

Metrics that measure training behavior and output quality are used to judge RLHF pipelines. Mean reward shows how well the generated answers match up with what the reward model expects, and reward standard deviation shows how consistent the responses are. Policy loss, value loss, and total loss all give us information about how to make the model work better and make it more in line with the reward model. Entropy ensures a balance between exploration and exploitation. PPO returns, which represent cumulative discounted rewards, are monitored to track alignment improvements. Textual measures, like the average length of generated tokens, show how output behavior changes. By keeping an eye on these measures, we can be sure that aligning rewards doesn't introduce any unintended biases, which keeps the quality and consistency of responses.

IV. RESULTS AND DISCUSSION

The results and discussion may be combined into a common section or obtainable separately. They may also be broken into subsets with short, revealing captions. An easy way to comply with the conference paper formatting requirements is to use this document as a template and simply type your text into it. This section should be typed in character size 10pt Times New Roman. The test results give a thorough look at how

e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:07/Issue:01/January-2025 Impact Factor- 8.187 www.irjmets.com

Reinforcement Learning from Human Feedback (RLHF) makes large language models (LLMs) better at specific tasks by improving their quality and alignment. By combining Proximal Policy Optimization (PPO) with a reward model that has been trained to reflect human preferences, RLHF makes it possible for models to produce results that are more in line with what experts expect and what users want. RLHF always led generative models to make responses that were correct, consistent, and relevant to the environment in both architectures. Because reinforcement learning is stochastic in nature, metrics were recorded at each PPO step instead of at longer epoch-level breaks. Early data showed a lot of variation, which was normal for the model's experimental phase. Over time, metrics like mean reward, policy loss, value loss, and return predictions showed steady trends toward greater stability and alignment with the reward model's standards. These trends show that the RLHF model's policy was getting better and better over time.

Experimental Setup

To make sure the pairwise dialogue dataset had high-quality inputs, it was preprocessed. It was used to train the Roberta-GPT-2, and Llama based RLHF architectures by comparing accepted and refused responses pairwise. The Roberta reward model had a batch size of 16, but the Llama reward model, which was bigger, had a batch size of 2 with 8 gradient accumulation steps to make up for the lack of resources. The RLHF training went on for 1000 steps, with 16 steps per batch and a learning rate of 1.41e-4. During the whole process, metrics like reward mean, policy loss, entropy, and generated answer length were monitored. LoRA and quantization methods were used for efficient training and resource usage, especially for the Llama process because it has a big model.

Results Of Rlhf Pipeline With Roberta Reward And Gpt-2 Generative Model

The Roberta-based reward model led the GPT-2 generative model through PPO in this case. The Mean Reward metric (Figure 5) steadily went up from about 0.18 to 0.49, showing that the model was getting better at producing answers that met the standards of the reward model. Early changes in the Standard Deviation of Reward metric showed the exploratory behavior of the model, but these changes dropped over time, showing that the quality of the responses was becoming more consistent. The Total Loss metric (Figure 6) went down a lot, which shows that the parameters were optimized well. At first, high entropy values (Figure 6) showed a lot of different exploratory outputs. As training went on, these answers became more focused. Temporary increases in entropy later in training suggested controlled exploratory behavior for further refinement. Response Length metrics (Figure 8) were unstable at first but then became stable, which shows that the model updated to generate more complex answers that are relevant to the situation.

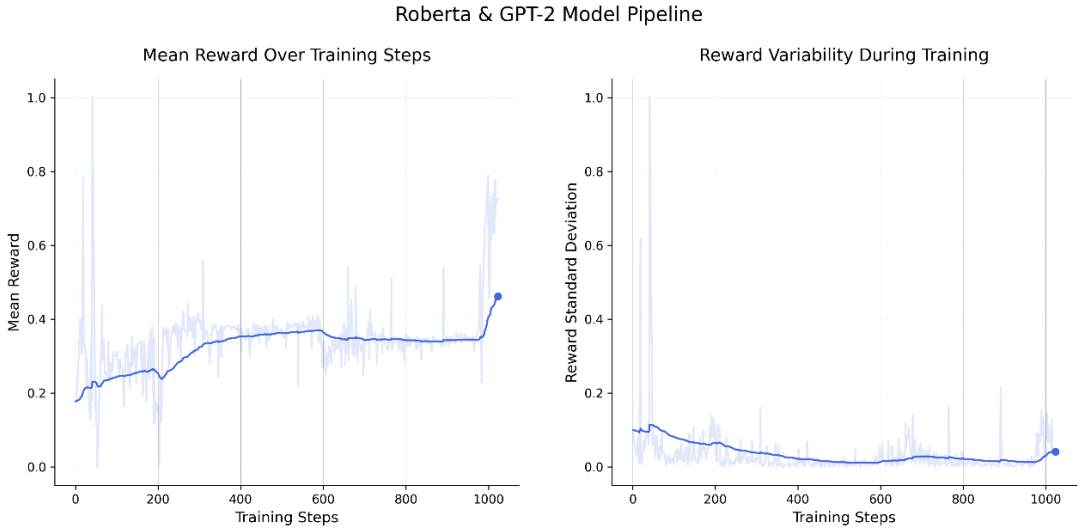

Figure 5: Mean and Standard Deviation of Reward for Roberta-GPT-2 RLHF Model

e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:07/Issue:01/January-2025 Impact Factor- 8.187 www.irjmets.com

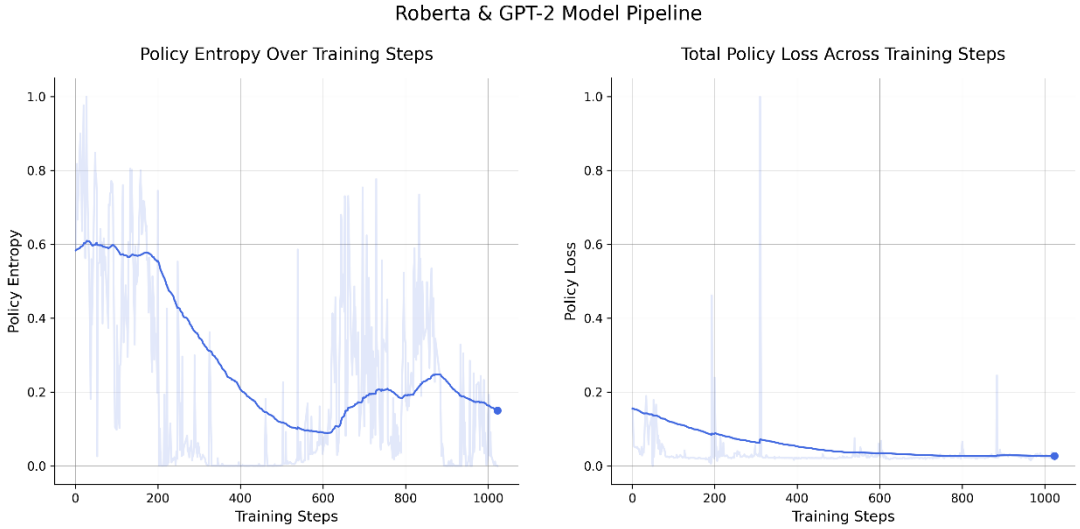

Figure 6: Policy Entropy and Total Loss for Roberta-GPT-2 RLHF Model

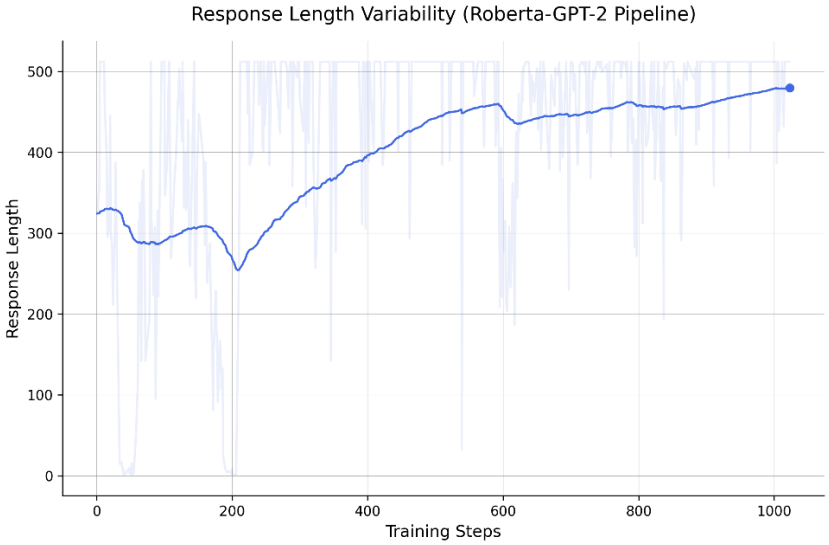

Figure 8: Response Lengths from Roberta-GPT-2 RLHF Model

Results Of Rlhf Pipeline With Llama Reward And Generative Model

The Llama-3.1 reward model (8B parameters) led the Llama-3.2 generative model (1B parameters) in this setup. The Mean reward metric (Figure 9) showed steady growth, going from about 0.4 to 0.79, showing that the model was becoming more in line with reward criteria. The Reward Standard Deviation metric slowly became less variable, which means that results were more consistent. The Total Loss measure (Figure 10) went down by a lot, which means that the strategy was optimized well. Over time, entropy values went from being high at first to being lower over time (Figure 10), showing a change from exploratory to more focused and improved policy behavior. The model's ability to balance length and clarity in answer generation is shown by the steadiness in answer Length measures (Figure 11).

e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:07/Issue:01/January-2025 Impact Factor- 8.187 www.irjmets.com

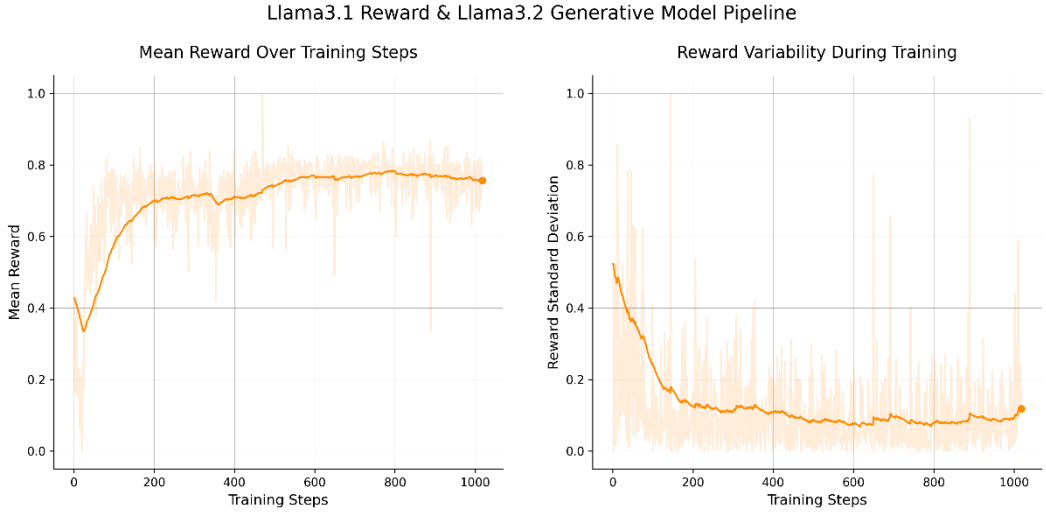

Figure 9: Mean and Standard Deviation of Reward for Llama-based RLHF Model

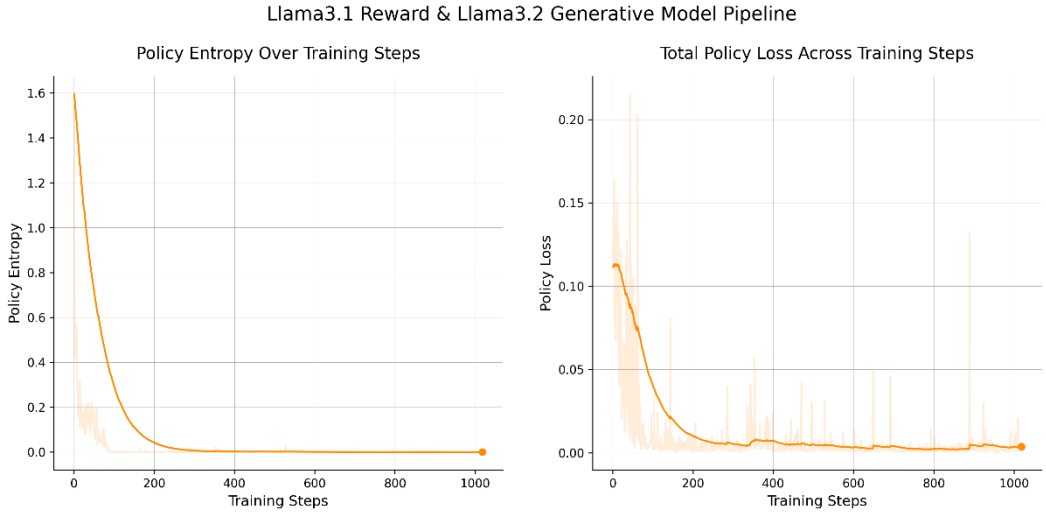

Figure 10: Policy Entropy and Total Loss for Llama-based RLHF Model

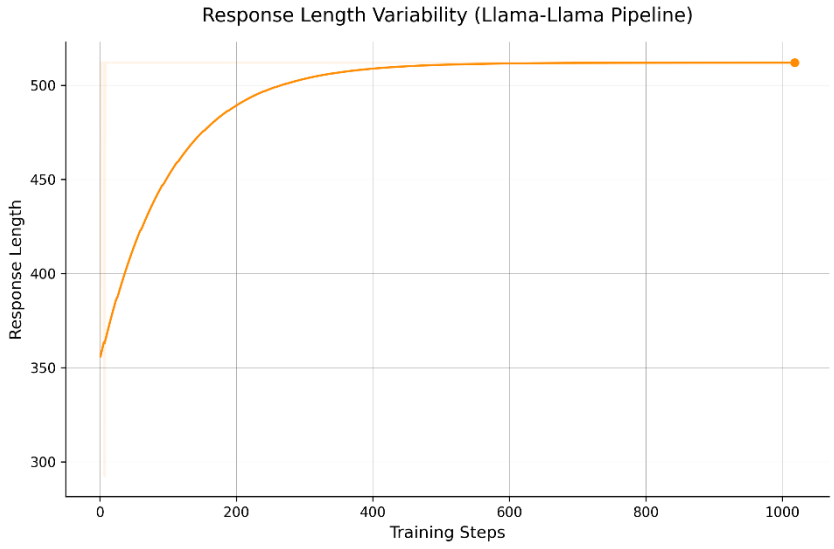

Figure 11: Response Lengths from Llama-based RLHF Model

Comparative Analysis Of Rlhf Pipelines

It was clear that both RLHF pipelines became more stable and oriented better. How they were trained, though, showed that they were not at all the same. The Llama-based system had more steady and smooth rises in Mean

e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:07/Issue:01/January-2025 Impact Factor- 8.187 www.irjmets.com

Reward and steady falls in Reward Standard Deviation. This demonstrated actions that were steady and of high quality. The Roberta-GPT-2 system was less stable at first and needed more training steps to become steady. Total Loss and entropy measurements made it clear that the Llama-Llama pipeline did a better job of balancing exploration and exploitation. The Llama process was better at keeping answer lengths steady, but the results from both could change and get more complicated. The system worked better, training went more easily, and changes were more stable when it was based on Llama.

Table 1. Comparative Analysis of RLHF Pipelines

Metric	Roberta + GPT-2 Pipeline		Llama + Llama Pipeline	
Training Steps	0	1000	0	1000
Mean Reward	0.177	0.498	0.428	**0.786**
Reward Standard Deviation	0.10	0.086	0.523	**0.173**
Total Policy Loss	0.155	0.022	0.115	**0.0013**
PPO Variance in Returns	0.413	0.088	0.151	**0.048**
Policy Entropy	0.583	0.002	1.593	**1.631e-06**
Value Function Error	7.156	0.152	2.23	**0.02**
Responses Length	324 Tokens	445 Tokens	356 Tokens	**512 Tokens**

V. CONCLUSION

This research looked at how Reinforcement Learning from Human Feedback (RLHF) can be used to improve the alignment and usefulness of large language models (LLMs) in medical discussion. By adding a reward model that has been trained to understand human preferences to a Proximal Policy Optimization (PPO) loop, the study showed big improvements in getting results that make sense in the given context and are medically correct. If we look at the Roberta-GPT-2 RLHF pipeline and the Llama3.1-8B - Llama3.2-1B RLHF pipeline side by side, the Llama-based design comes out on top. The Llama-based system gave better results because it had a bigger model and understood the context better. The Roberta-GPT-2 pipeline worked better, but it had more gaps, so it needed more training. Even though these results are good, there are still challenges. It's still hard to work with RLHF because it takes a lot of computer power. Adding more criteria to reward systems that look at things like scientific correctness, moral thinking, and cultural awareness could help align models even more and produce better results. These models could be more useful and adaptable if they could be used in other areas, like the school or law systems. Real-time changes to reward models and methods that put a human in the loop are two more ways that standards can be kept up to date. It's important to solve problems with model interpretability and tokenization limits so that users trust LLMs and they can handle longer and more complicated exchanges well.

VI. REFERENCES

[1] Sutton, R. S., & Barto, A. G. (1998). Reinforcement Learning: An Introduction. Cambridge, MA: MIT Press.

[2] Minaee, S., Mikolov, T., Nikzad, N., Chenaghlu, M., Socher, R., Amatriain, X., & Gao, J. (2024). Large Language Models: A Survey. [Journal/Publisher details, if available].

[3] Balne, C. C. S., Bhaduri, S., Roy, T., Jain, V., & Chadha, A. (2024). Parameter Efficient Fine Tuning: A Comprehensive Analysis Across Applications. University of Southern California; Amazon; Deepsig Inc.; Stanford University.

[4] Ding, N., Qin, Y., Yang, G., Wei, F., Yang, Z., Su, Y., et al. (2023). Parameter-efficient fine-tuning of large-scale pre-trained language models. Nature Machine Intelligence, 5(3), 220–235. https://doi.org/10.1038/s42256-023-00626-4.

[5] Dettmers, T., Pagnoni, A., Holtzman, A., & Zettlemoyer, L. (2023). QLoRA: Efficient Finetuning of Quantized LLMs. Preprint. https://arxiv.org/abs/2305.14314.

[6] Nagel, M., Fournarakis, M., Amjad, R. A., Bondarenko, Y., van Baalen, M., & Blankevoort, T. (2021). A White Paper on Neural Network Quantization. Qualcomm AI Research.

https://arxiv.org/abs/2106.08295.

[7] Mahan, D., Phung, D. V., Rafailov, R., Blagden, C., Lile, N., Castricato, L., Franken, J., Finn, C., & Albalak, A. (2024). Generative Reward Models. Preprint. https://arxiv.org/abs/2410.12832.

[8] Liu, C. Y., Zeng, L., Liu, J., Yan, R., He, J., Wang, C., Yan, S., Liu, Y., & Zhou, Y. (2024). Skywork-Reward: Bag of Tricks for Reward Modeling in LLMs. Technical Report. Skywork AI, Kunlun Inc. https://arxiv.org/abs/2410.18451.

[9] Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C. L., Mishkin, P., ... & Lowe, R. (2022). Training language models to follow instructions with human feedback. OpenAI. https://arxiv.org/abs/2203.02155.

[10] Ziegler, D. M., Stiennon, N., Wu, J., Brown, T. B., Radford, A., Amodei, D., Christiano, P., & Irving, G. (2020). Fine-Tuning Language Models from Human Preferences. arXiv. https://arxiv.org/abs/1909.08593.

[11] Schulman, J., Wolski, F., Dhariwal, P., Radford, A., & Klimov, O. (2017). Proximal Policy Optimization Algorithms. ArXiv. https://arxiv.org/abs/1707.06347

[12] Wu, T., Zhu, B., Zhang, R., Wen, Z., Ramchandran, K., & Jiao, J. (2023). Pairwise Proximal Policy Optimization: Harnessing Relative Feedback for LLM Alignment. ArXiv. https://arxiv.org/abs/2310.00212

[13] Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., & Sutskever, I. (2019). Language models are unsupervised multitask learners. OpenAI. https://openai.com/research/language-models.

[14] Touvron, H., Martin, L., Stone, K., et al. (2023). Llama 2: Open Foundation and Fine-Tuned Chat Models. Meta AI. https://arxiv.org/abs/2307.09288.

[15] Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., & Stoyanov, V. (2019). RoBERTa: A Robustly Optimized BERT Pretraining Approach. https://arxiv.org/abs/1907.11692.

[16] Stiennon, N., Ouyang, L., Wu, J., Ziegler, D. M., Lowe, R., Voss, C., Radford, A., Amodei, D., & Christiano, P. (2020). Learning to summarize from human feedback. ArXiv. https://arxiv.org/abs/2009.01325