Information Dense Question Answering for RLHF

Stanford CS224N Custom Project

Chethan Bhateja
Department of Computer Science
Stanford University
chethanb@stanford.edu

Abstract

LLMs trained with RLHF exhibit a bias toward longer responses, leading to repetition and answers that are tedious to parse. The goal of this project is to improve the conciseness of LLMs trained with RLHF for question answering. Inspired by information theory concepts, I examine whether modifying rewards based on token entropy, token similarity, and length of responses during RLHF increases information density of downstream generated responses. Though some my interventions are able to decrease length and repetitiveness and improve information density, these come at the cost of potentially worse response quality compared to ground truth human responses.

1 Key Information to include

- Mentor: Kaylee Burns
- External Collaborators: None
- Sharing project: No

2 Introduction

My goal was to increase the information density of an LLM's responses for question answering. I became interested in this problem because I've noticed most text I read can be shortened substantially by changing phrases or sentence structure with almost no loss of information. Large language models, particularly those trained with RLHF, are biased even further towards longer outputs, likely due to humans ranking these outputs as more favorable (Shen et al., 2023). For example, I asked ChatGPT "How long past the expiration can toothpaste last?" and it gave a long 3 paragraph response, which I got it to shorten to the following:

> Toothpaste can often be used a few months beyond its expiration date, but it's best to replace it if you notice changes in color, texture, or smell. Factors like storage conditions can affect its stability. While using toothpaste past the expiration date might not be harmful, it's crucial to prioritize oral hygiene and replace it if there are any doubts. If in doubt, consult with a dentist.

I came up with the much shorter summary below, which contains the key information:

> You can often use toothpaste a few months past its expiration, but toss it if you see changes. Storage conditions impact stability. Prioritize hygiene and ask a dentist if unsure.

Prior approaches include length penalties during RLHF (Singhal et al., 2023), altered token selection during decoding (Xu et al., 2023), or training a product of experts model to isolate simple biases

(Shen et al., 2023), but these approaches are either unsuccessful or don't truly optimize my desired objective. There are ways of quantifying information density and repetitiveness, such as entropy and KL divergence, and my approach in this project was to apply information theoretic concepts during RLHF in an attempt to increase information density of responses. I found these approaches to some benefits in decreased length and increased information density, but likely at the cost of response quality.

3 Related Work

Prior approaches relevant to my goal explore other ways of changing reward during RLHF, training a Product of Experts model during RLHF to isolate simple biases, or altering the way tokens are chosen during decoding.

Standard RLHF fine-tuning begins with a supervised fine-tuned (SFT) model π^{SFT}, trains a reward model R_θ on context response pairs (x, y), and optimizes π_ϕ on the rewards $R_\theta(x, y)$ via the PPO(Schulman et al., 2017) objective. In the direction of altering rewards during RLHF, Singhal et al. (2023) explore adding a length penalty on the responses y to the rewards or omitting tokens after a length threshold from PPO training. Though these interventions decreased length compared to standard RLHF, they found both interventions still had longer response lengths than the base SFT model and achieved lower reward (Singhal et al., 2023). I also find this reward objective a bit shallow compared to information density, as a short piece of text can still be uninformative and optimizing information density instead could lower length naturally.

Shen et al. (2023) take an alternative approach by training a Product of Experts with a small model and a larger model during the reward modeling stage, calling the small model a bias LLM. They argue that the simpler bias LLM learns simple and undesirable features of responses such as length and therefore remove the bias LLM during RLHFShen et al. (2023). While Shen et al. (2023) do show some reduction in length and improvements in other metrics, I find their approach somewhat unsatisfying as it is unclear what objective it optimizes and removing the bias LLM could also remove desirable features of responses.

Finally, Xu et al. (2023) examine improving informativeness in open-ended neural text generation, where a language model is given a context $\{x_1, \ldots, x_m\}$ and must output an n token continuation. Xu et al. (2023) identify the competing objectives of avoiding repetition and avoiding incoherence, where a model can drift off topic when repetition is penalized heavily. They propose **look-back decoding**, a method of token selection that avoids low KL divergences between generated tokens while bounding KL divergence from the reference prompt, which they find avoids repetition while preserving coherenceXu et al. (2023). I found the idea of looking at token distributions rather than tokens themselves quite clever. However, I also feel the objective of avoiding repetition is slightly undesirable compared to information density, as text can avoid being repetitive while still being verbose or uninformative. The fact that the algorithm is applied at decoding time has the advantage of avoiding retraining, which is a huge benefit, but also means that the LLM is not truly optimized for the desired objective, which could lead to sub-optimal outputs and slowdowns at test time.

4 Approach

RLHF fine-tuning begins with a supervised fine-tuned (SFT) model π^{SFT}, trains a reward model R_θ on preferences $y^1 \succ y^2$ between pairs of responses (y^1, y^2) for a given context x, and optimizes π_ϕ via the PPO objective

$$\mathbb{E}_{x\sim D}[\mathbb{E}_{y\sim\pi_\phi(x)}[R_\theta(x, y) - \lambda \log(\pi_\phi(y|x)/\pi^{SFT}(y|x)]]$$

where the KL-divergence penalty avoids over-fitting on rewards by constraining the policy to be close to the supervised model.

My approaches fall into the category of altering rewards during RLHF, since RLHF is a standard way of training a language model to obey certain preferences. Building off of Singhal et al. (2023)'s repository for examining length biases in RLHF, I examined adding the following rewards to the reward model during PPO training.

4.1 Token Entropy Bonus

During PPO fine-tuning I was able to get the log probabilities of each generated token $\log p(y_1), \ldots, \log p(y_n)$ from the language model. Then for each response I calculated the entropy

$$H(y) = \sum_i p(y_i) \log p(y_i)$$

Intuitively, this objective makes token probabilities slightly more uniform which can increase the uncertainty, and therefore the information density, of responses. However, it does have the drawback of potentially generating sub-optimal responses with respect to the original rewards or injecting undesired randomness into responses.

4.2 Token Pairwise KL Divergence Bonus

(Xu et al., 2023) used KL divergences of generated tokens to great effect in Lookback Decoding, finding that KL divergences between token distributions reflected token similarity quite well as shown in the KL heat map from their paper in Figure 1.

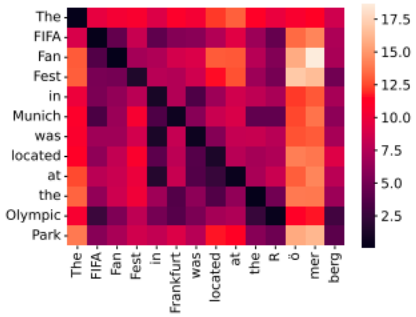

Figure 1: KL divergence heat map between two sentences from Xu et al. (2023)

In the heat map, similar words have lower KL divergences, which suggests that comparing through KL divergences provides a more powerful information theoretic notion of similarity than simply checking if tokens are exactly the same. This gave me the idea to use KL divergences during RLHF instead of decoding to train the model to avoid repetitions at this stage. Therefore, I added the following bonus to incentivize higher KL divergences between generated response tokens:

$$\sum_{i,j} KL(p(y_i), p(y_j))$$

Xu et al. (2023) used this objective over a fixed backward window when choosing tokens during decoding and additionally required some coherence to the context. In contrast, implementing this during RLHF avoids the concern of over optimizing and results in a model that is truly optimized with the desired objective of avoiding repetition.

4.3 English Length Penalty

Finally, I noticed that Singhal et al. (2023) used a penalty on token lengths of responses, but the lengths of responses in English can be quite different, and this matter significantly for humans. In particular, lower entropy English words such as "the" are naturally shorter, while such words can have the same length in token space. This gave me the idea to put a penalty on English length instead of using token length as in Singhal et al. (2023). Conretely, as in Singhal et al. (2023), I use the penalty

$$\sigma(1 - \frac{\text{len}(y)}{N})$$

where N is a maximum length we do not want to exceed but $\text{len}(y)$ is now English length instead of token length.

4.4 Models

I primarily used the Llama-7B (Touvron et al., 2023) and Pythia-1B (Biderman et al., 2023) models from Hugging Face in my experiments for both training reward models and language generation during PPO. However, I found PPO training using the Pythia-1B reward model to be unstable. I also attempted to use Google's Gemma model but ran into some errors which I couldn't resolve easily.

4.5 Baselines

For my baselines, I compare with both supervised fine-tuned (SFT) models from before RLHF and models trained with standard RLHF. For SFT models, I used the Llama-7B checkpoint from AlpacaFarm (Dubois et al., 2024) and the Pythia-1B model (Biderman et al., 2023) from Hugging Face. For RLHF, I used Singhal et al. (2023)'s repository to train a reward model from each of these SFT models and then fine-tuned with the standard rewards and no additions.

5 Experiments

5.1 Data

As in Singhal et al. (2023), I use the WebGPT dataset(Nakano et al., 2022), where each example features a question, context the WebGPT model found while browsing for the question, and two answers, each with a score in $[-1, 1]$ denoting the strength of the preference for that answer over the other answer. This preference data is used during RLHF to train the reward model, before then running PPO on the reward model so that downstream language generation reflects the desired preferences.

5.2 Evaluation method

I use the same metrics used in Lookback Decoding (Xu et al., 2023). First, I use the **repetition** and **diversity** metrics from Su et al. (2022) to get some measure of the repetitiveness and information density of my generated responses. For a generated piece of text, Su et al. (2022) define the n-gram level repetition, a score in $[0, 100]$, as

$$\textbf{rep-n}(\hat{x}) = 100 \left(1 - \frac{|\text{unique n-grams}(\hat{x})}{\text{total n-grams}(\hat{x})}\right)$$

This measures the proportion of duplicated n-grams in a piece of text. Su et al. (2022) then define

$$\textbf{diversity}(\hat{x}) = \prod_{n=2}^{4} \left(1 - \frac{\textbf{rep-n}(\hat{x})}{100}\right)$$

To evaluate the quality of responses, I use the **MAUVE** metric, which clusters generated and golden human outputs in an embedding space and compares them using KL divergences (Pillutla et al., 2021). MAUVE scores are in $[0, 1]$ with higher MAUVEs being closer to human ground truth text. After completion of RLHF, I selected 1000 examples from the WebGPT dataset and ran MAUVE on these to compare my generated model outputs to the ground truth human text.

5.3 Experimental details

I took around a day to train the reward model for Llama-7B and half a day to train a reward model for Pythia-1B. Then I ran PPO fine-tuning for 500 epochs with Pythia, which took 8 hours per run, and 250 epochs for Llama, which took about 8 hours per run. I used learning rates of 1e-5.

5.4 Results

	Mauve	Repetition-3	Diversity	Length
RLHF + Entropy	0.175	0.155	0.990	199
RLHF + Pairwise KL	0.376	0.561	0.969	385
RLHF + Len Penalty	0.117	0.100	0.994	145
RLHF	0.287	0.239	0.983	380
SFT	0.337	0.174	0.984	441

Table 1: Llama-7B Results

Evaluation metrics with the Llama-7B model are shown in 1. We can see that MAUVE and repetition appear to compete with each other, which makes sense. The entropy bonus and length penalty reduce repetition by 35% and 58% respectively compared to standard RLHF, potentially helped by their shorter length. However, they also significantly decrease MAUVE with human ground truth text compared to standard RLHF. The authors of MAUVE mention that MAUVE may not necessarily indicate the true quality of output text since it only compares to the distribution of human text but this metric still suggests that the quality of our output may be worse. It is interesting to note that as we were hoping, incentivizing, entropy does seem to naturally shorten length, leading to more concise responses and high diversity.
The results of the pairwise KL divergence bonus are surprising. Repetition scores are out of 100, meaning that the increase in repetition may not be substantial, but I was expecting incentivizing high KL divergences between tokens to have the opposite effect of lowering repetition. Further surprising me is the fairly substantial improvement in MAUVE compared to SFT and RLHF, which I have not been able to explain.

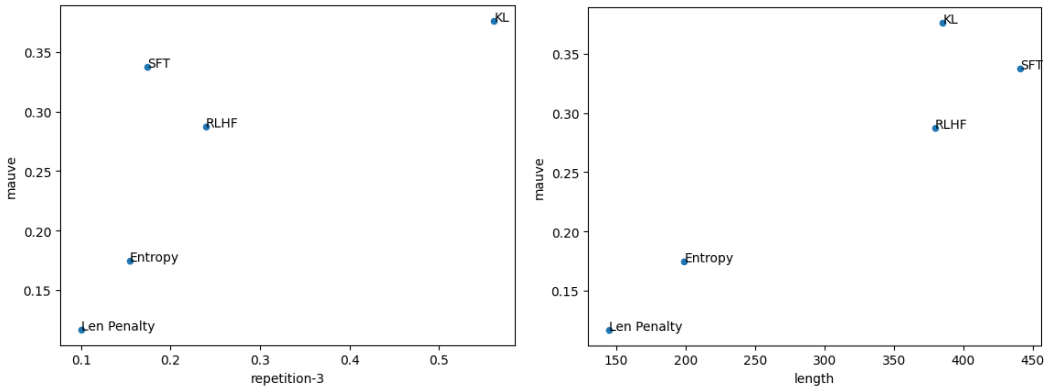

Figure 2: MAUVE vs. Repetition and Length

To dig deeper into the relationship between repetition and MAUVE, we plot repetition vs. MAUVE in Figure 2. Unfortunately if we trust the MAUVE metric, these Pareto frontiers suggests that our interventions may be harmful with respect to quality but examples in our analysis section suggest that all answers are reasonably well-formed. Time sensitive readers may prefer the substantially shorter responses from our interventions over those from standard SFT and RLHF. It would also be interesting to see if there is a way to smoothly trade of these factors, perhaps through reweighting rewards.

	Mauve	Repetition-3	Diversity	Length
RLHF + Entropy	0.013	1.367	0.779	1243
RLHF + Pairwise KL	0.013	1.037	0.792	1254
RLHF + English Penalty	0.011	0.596	0.859	1269
RLHF + Len Penalty	0.019	5.170	0.771	755
RLHF	0.016	2.339	0.783	1248
SFT	0.019	2.921	0.521	1160

Table 2: Pythia-1B Results

I found RLHF with Pythia-1B to be unstable, with responses becoming gibberish repeating characters at 100 epochs, which I demonstrate may be due to reward modeling in the analysis section. Nevertheless, my results with the Pythia-1B model with 50 epochs of PPO training are shown in 2. The MAUVE metrics and repetition metrics are substantially poorer for Pythia-1B than for Llama-7B. However, looking at the relative results, we do see a substantially lower repetition and higher diversity with each of our 3 interventions, particularly with the penalty on length in English. The MAUVE metrics are slightly lower than RLHF but still in the same ballpark, so perhaps the increased information density of our answers makes the tradeoff worthwhile here.

6 Analysis

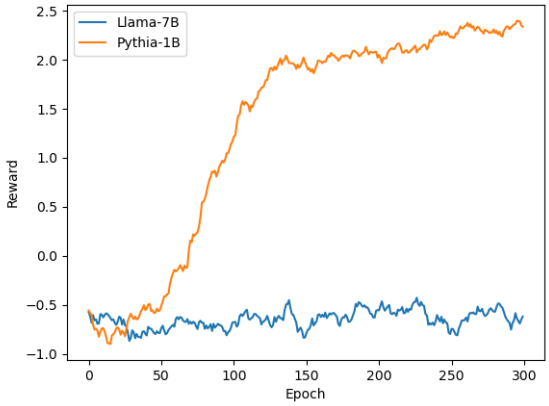

Figure 3: Rewards of Llama-7B and Pythia-1B reward models during standard RLHF

For one clue as to why Pythia-1B performance is poor compared to Llama-7B, I compared the reward between Llama-7B and Pythia-1B for during standard RLHF in Figure 3. We can see that while the rewards from the Llama-7B reward model stay roughly constant during RLHF, the Pythia-1B reward model shows substantial increases in reward quite early in training, right around 50 iterations where I observed the response quality degrading. Perhaps the Pythia-1B reward model can simply be exploited too easily, and we must use larger models for truly robust RLHF. Alternative methods without a reward model such as Direct Preference Optimization may be able to sidestep this issue (Rafailov et al., 2023).

Qualitatively, I found response length and tone very different among the methods. One example of a question with corresponding outputs is shown below.

Question	**We have a circulatory system, nervous system, and a great deal of organs, but what is responsible for actually generating heat? Our blood is warm, but what is heating it?**
RLHF	Our body produces heat through metabolism - the process by which we convert food into energy to use in our cells. This energy produced during metabolism can be measured as heat, and so this is why our body's temperature increases when its using energy!
SFT	Our body produces heat by metabolism - the process in which your cells use energy to carry out the tasks needed to survive. During this process, molecules collide with each other, creating friction which produces heat. The greater the number of collisions between molecules, the higher the temperature will be.
Entropy Bonus	Our body produces heat through metabolism - the process by which cells use energy to carry out biochemical processes in the body.
KL Bonus	Our body produces heat through cellular respiration which involves the breakdown of glucose in the cells to use as energy. This process generates ATP (Adenosine Triphosphate), which is used by the body's tissues to carry out the needed functions. During this reaction, some heat is produced and released into the surrounding environment, helping maintain the body's temperature at a normal level.
Length Penalty	Our body produces heat through metabolism.

Upon looking at these responses, all of them seem to be well formed so perhaps the degradation of the MAUVE metric is not of too great concern. Among the baselines we can see that RLHF has repeats "energy" several times ("into energy", "this energy", "using energy") and gives a detailed somewhat enthusiastic response, while the SFT model gives a good if slightly long answer with a bit of repetition. Compared to these two answers, the model trained with a bonus on KL divergences gives a very informative answer which I like a lot, describing the actual scientific processes that are occurring without too much repetition. Among the shorter answers, the model with entropy bonus gives a concise answer containing the main points while the model with the length penalty gives an answer that is probably too short and would demand some follow-up questions.

7 Conclusion

Through this project, I found that modifying rewards during RLHF training with information theoretic measures and length penalties can lead to a variety of different response modalities, which may appeal to different types of users. I discovered the benfeits of drawbacks of using certain strategies, such as entropy, kl divergence of tokens, and length penalties, as well of some of the challenges of the RLHF workflow. However, I think a more powerful approach could allow us to span the full Pareto frontier of tradeoffs between length and response quality. I can imagine a user-interface for a LLM where a user might drag certain sliders for response length or repetitiveness and the LLM could respond to these. RLHF is also inflexible in having to have a separate reward model for each of these choices - perhaps there is a way to train a reward model conditioned on length or other user preferences. Finally, I feel a better approach might be to modify direct preference optimization instead of using RLHF with a reward model and PPO, which would could avoid instability of training from reward models.

References

Stella Biderman, Hailey Schoelkopf, Quentin Anthony, Herbie Bradley, Kyle O'Brien, Eric Hallahan, Mohammad Aflah Khan, Shivanshu Purohit, USVSN Sai Prashanth, Edward Raff, Aviya Skowron, Lintang Sutawika, and Oskar van der Wal. 2023. Pythia: A suite for analyzing large language models across training and scaling.

Yann Dubois, Xuechen Li, Rohan Taori, Tianyi Zhang, Ishaan Gulrajani, Jimmy Ba, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. 2024. Alpacafarm: A simulation framework for methods that learn from human feedback.

Reiichiro Nakano, Jacob Hilton, Suchir Balaji, Jeff Wu, Long Ouyang, Christina Kim, Christopher Hesse, Shantanu Jain, Vineet Kosaraju, William Saunders, Xu Jiang, Karl Cobbe, Tyna Eloundou, Gretchen Krueger, Kevin Button, Matthew Knight, Benjamin Chess, and John Schulman. 2022. Webgpt: Browser-assisted question-answering with human feedback.

Krishna Pillutla, Swabha Swayamdipta, Rowan Zellers, John Thickstun, Sean Welleck, Yejin Choi, and Zaid Harchaoui. 2021. Mauve: Measuring the gap between neural text and human text using divergence frontiers.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D. Manning, and Chelsea Finn. 2023. Direct preference optimization: Your language model is secretly a reward model.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal policy optimization algorithms.

Wei Shen, Rui Zheng, Wenyu Zhan, Jun Zhao, Shihan Dou, Tao Gui, Qi Zhang, and Xuanjing Huang. 2023. Loose lips sink ships: Mitigating length bias in reinforcement learning from human feedback.

Prasann Singhal, Tanya Goyal, Jiacheng Xu, and Greg Durrett. 2023. A long way to go: Investigating length correlations in rlhf.

Yixuan Su, Tian Lan, Yan Wang, Dani Yogatama, Lingpeng Kong, and Nigel Collier. 2022. A contrastive framework for neural text generation.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. Llama: Open and efficient foundation language models.

Nan Xu, Chunting Zhou, Asli Celikyilmaz, and Xuezhe Ma. 2023. Look-back decoding for open-ended text generation.

A Appendix (optional)

If you wish, you can include an appendix, which should be part of the main PDF, and does not count towards the 6-8 page limit. Appendices can be useful to supply extra details, examples, figures, results, visualizations, etc. that you couldn't fit into the main paper. However, your grader *does not* have to read your appendix, and you should assume that you will be graded based on the content of the main part of your paper only.