CSI -5130 Artificial Intelligence

Adaptive Prompt Optimization using Reinforcement Learning (RLHF-lite)

Agna Antony, Vineela Rao Akasapu, Mubassir Gamfoi

Abstract

In order to maximize the performance of Large Language Models (LLMs), prompt engineering is essential. In this project, a lightweight version of Reinforcement Learning from Human Feedback (RLHF-lite) is used to propose an Adaptive Prompt Optimization system. By iteratively creating, assessing, and updating prompt candidates, the system automatically enhances prompts for tasks like summarizing or answering questions. To direct the optimization process, we use automated evaluation metrics (ROUGE, BLEU) and reinforcement learning (RL) algorithms. The findings show how reinforcement learning can improve prompt quality, producing LLM outputs that are more precise and consistent over time.

1. Introduction

In natural language processing, large language models (LLMs) like GPT-4, Mistral, and T5 are now essential tools. However, the input prompt has a significant impact on LLM behavior. While a well-designed prompt can greatly increase clarity and correctness, a poorly phrased prompt may result in irrelevant, verbose, or inaccurate outputs. Manually designing prompts is subjective, extremely iterative, reliant on professional intuition and not adaptable to different tasks. As a result, automated prompt optimization is becoming a significant area of study. Reinforcement Learning from Human Feedback (RLHF) has been successful in aligning large-scale models, but it typically requires significant resources like human preference data, reward models, and substantial computation. To explore a

more accessible alternative, this project investigates RLHF-Lite, a simplified reinforcement learning approach that uses automated metrics instead of human feedback. By combining a lightweight LLM (FLAN-T5) with a Q-learning agent that selects from multiple prompt templates using ROUGE/BLEU-based rewards, this system demonstrates how a small-scale RL framework can adaptively improve prompt effectiveness.

The goal of this project is to use reinforcement learning to automate prompt optimization using a simplified methodology known as RLHF-lite. This method enables small-scale, experimental RL-based prompt engineering by using automated metrics to compute rewards instead of manually curated human feedback.

Large Language Models (LLMs) represent a major milestone in natural language processing, enabling machines to perform tasks such as summarization, reasoning, translation, and dialogue generation with human-like fluency. Modern LLMs including GPT-4, Mistral, LLaMA, and Google's T5 family are built on the Transformer architecture, which uses self-attention mechanisms to understand long-range dependencies in text. Instead of processing words sequentially, Transformers attend to all tokens simultaneously, allowing them to capture nuanced linguistic patterns and contextual relationships.

2. Background and Related Work

2.1 Large Language Models (LLMs)

A significant advancement in natural language processing, Large Language Models (LLMs) allow machines to carry out tasks like summarization, reasoning, translation, and dialogue generation with human-like fluency. The Transformer architecture, which employs self-attention mechanisms to comprehend long-range dependencies in text, is the foundation of contemporary LLMs, such as GPT-4, Mistral, LLaMA, and Google's T5 family. Transformers are able to capture subtle linguistic patterns and contextual relationships because they handle all tokens at once rather than processing words one after the other.

LLMs are sequence-to-sequence or autoregressive generators that use probability distributions to generate text one token at a time. Their performance

can be extremely sensitive to the input prompt's structure, clarity, and intent because of this design. Coherence, correctness, and generation quality can all be considerably changed by small phrasing adjustments. Consequently, prompt engineering has emerged as a crucial method for successfully directing LLM behavior.

2.2 Prompt Engineering

The process of creating and improving input instructions to elicit desired responses from LLMs is known as prompt engineering. To maximize model behavior, effective prompts must strike a balance between specificity, clarity, and stylistic cues. Studies have demonstrated that even minor changes, like substituting "TL;DR" for "Summarize the following text," can result in noticeably different results. Because LLMs primarily rely on learned patterns from extensive text corpora, where specific phrases serve as potent semantic triggers, this sensitivity results.

Typical prompt templates consist of:

"Give a brief explanation: (promotes short explanations)

"TL;DR: (creates short, straightforward summaries)

"Summarize in a single sentence: (forces information to be compressed)

"What are the main points?" (produces structured or bullet-style answers)

2.3 Language Model Reinforcement Learning

A potent paradigm for matching LLM behavior with human preferences is Reinforcement Learning (RL). Through trial-and-error interactions with an environment, reinforcement learning (RL) allows the model (or agent) to improve iteratively, in contrast to supervised fine-tuning. In the context of LLMs:

State: the text or context at hand

Action: the result or decision made by the model (e.g., choosing a prompt, generating text)

Reward: quality evaluation feedback (automated score, human preference)

2.3.1 RLHF-Lite for Prompt Optimization

To make reinforcement learning feasible in a lightweight and resource-limited environment, this project adopts a simplified reinforcement learning framework known as RLHF-Lite. Instead of relying on human-annotated preference data or large reward models, the system uses a minimal setup where the action corresponds to selecting a prompt template, and the environment is represented by the FLAN-T5-base model generating a summary in response to that prompt. The reward is computed automatically using ROUGE-1 and BLEU scores, providing an objective measure of how closely the generated summary aligns with the reference text. A standard Q-learning update rule is then applied to adjust the estimated value of each prompt template based on observed performance. This lightweight approach removes the need for human annotators, large-scale reward frameworks, and expensive computational resources. Instead, it focuses on adaptive prompt selection, demonstrating how reinforcement learning can effectively guide and improve LLM behavior even in constrained environments such as Google Colab.

3. System Design

The overall system is composed of three major components that interact to perform automated prompt optimization: the Target LLM, the Reward Engine, and the Prompt Optimizer. The Target LLM used in this project is FLAN-T5-base, a lightweight but instruction-tuned model capable of generating high-quality summaries from textual input. This model acts as the environment in which the reinforcement learning agent operates. The second component, the Reward Engine, evaluates each generated summary using standard automatic evaluation metrics primarily ROUGE-1 and BLEU. These scores are averaged to produce a single numerical reward signal that reflects how closely the model's output matches the reference summary. The third component is the Prompt Optimizer, implemented as a Q-learning agent. Its role is to select which prompt template to use for each iteration, observe the reward returned by the LLM, and update its internal Q-values accordingly. Over time, the agent learns which templates lead to higher-quality summaries and preferentially selects them.

To ensure diversity in the agent's action space, we defined a set of five prompt templates that represent different summarization styles: *"Summarize the following text:"*, *"TL;DR:"*, *"Explain this briefly:"*, *"What are the key points?"*, and *"Rewrite this concisely:"*. These templates capture varying levels of explicitness, conciseness, and stylistic framing, enabling the RL agent to explore which forms of instruction yield the best summarization results when paired with FLAN-T5.

The reinforcement learning setup follows a standard Q-learning formulation. Since the task does not require explicit state modeling, the state is treated as implicit, and the action corresponds to selecting one of the available prompt templates. The reward is computed using the average of ROUGE-1 and BLEU scores as shown below:

$$reward = (ROUGE1 + BLEU) / 2$$

Q-value updates follow the traditional update rule:

$$Q[a] = Q[a] + \alpha * (reward - Q[a])$$

where the learning rate is set to $\alpha = 0.2$. Exploration is controlled through an epsilon-greedy strategy, using $\epsilon = 0.5$ to encourage the agent to explore different templates during training. This system design provides a simple yet effective framework for demonstrating how reinforcement learning can adaptively improve prompt selection for LLM-based summarization tasks.

4. Implementation Details

4.1 Frameworks Used

The implementation of this project was carried out in Python using several well-established libraries commonly used in natural language processing and machine learning research. The Transformers library from HuggingFace was used to load and run the FLAN-T5-base model, providing the tools necessary for tokenization, model inference, and configuration. The Evaluate library was used to compute ROUGE-1 and BLEU scores, which served as the automated reward metrics in the RL training loop. Additional support libraries included NumPy for

numerical operations, random sampling, and maintaining Q-value structures, and Matplotlib for visualizing reward progress and Q-value evolution across training iterations. All experiments were executed in Google Colab, taking advantage of its GPU support to efficiently generate summaries during training.

4.2 Model

The target language model used in this project was FLAN-T5-base, accessed through the HuggingFace Transformers library using the model identifier "google/flan-t5-base". FLAN-T5-base contains approximately 250 million parameters and is instruction-tuned, meaning it responds well to explicit prompts and directive phrases. Its relatively compact size makes it suitable for experimentation within Colab's resource limits while still delivering strong summarization performance. This balance of efficiency and capability makes FLAN-T5-base an ideal model for studying how different prompt templates influence output quality during reinforcement learning.

4.3 Training Data

To train the reinforcement learning agent, a small handcrafted dataset consisting of five text passages paired with reference summaries was created. These examples were chosen to be diverse enough to test summarization behavior while keeping the system computationally efficient. Each training iteration sampled one pair at random, allowing the RL agent to interact with the model repeatedly over the same set of examples, assessing how different prompts affect summarization quality. Although small, this dataset was sufficient to demonstrate how Q-learning can adaptively improve prompt selection.

4.4 Training Loop

The training loop followed a structured reinforcement learning process repeated for 50 to 200 iterations, depending on the experiment. In each iteration, the Q-learning agent selected one of the five prompt templates using an ε-greedy strategy, where ε = 0.5 encouraged significant exploration of different templates. The selected

template was appended with a newline and the input text to form the full prompt provided to FLAN-T5-base, which then generated a summary. The generated output was evaluated by computing ROUGE-1 and BLEU scores, and the average of these scores served as the reward signal. This reward was used to update the Q-value of the selected prompt template through the standard Q-learning rule, allowing the agent to gradually learn which templates consistently produced higher-quality summaries. Over time, this training process enabled the agent to identify better-performing prompts, as reflected in improved reward values and template selection behavior.

5. Results

5.1 Input 3 - Spacecraft Passage (Before vs After)

The spacecraft passage showed the most significant improvement after reinforcement learning and serves as the strongest demonstration of the system's capabilities. For the input text *"The spacecraft drifted silently through the empty darkness of space,"* the baseline prompt *"Summarize the following text:"* produced the output *"The spacecraft sped past the horizon."* This summary deviated notably from the original meaning, introducing new imagery that was not present in the passage, and therefore received a relatively low reward score of 0.2316. After training, the Q-learning agent identified "TL;DR:" as the optimal prompt template. Using this template, the model generated *"The spacecraft drifted silently through the empty darkness of space,"* which closely matched the original text and resulted in a perfect reward score of 1.0. This example illustrates that the RL agent successfully learned to prefer the prompt that yielded the highest quality output under the ROUGE/BLEU-based reward system.

5.2 Interpretation of Results

The qualitative and quantitative differences observed in Input 3 provide strong evidence that the RLHF-Lite framework effectively learned to optimize prompt selection. The optimized prompt led to significantly better alignment with the input passage, demonstrating that reinforcement learning can guide the model toward more faithful summarization behavior. The improvement from a low reward to a

perfect score reflects the agent's ability to adjust its Q-values based on the performance of different templates, ultimately converging on the instruction that consistently produced the best outcomes. This result confirms the value of incorporating automated feedback within a lightweight RL paradigm to enhance language model performance.

5.3 Summary of All Three Examples

Although Input 3 produced the largest improvement, the other two evaluation examples also revealed informative trends. In the first input involving the well-known "quick brown fox" sentence, both the baseline and optimized prompts generated short paraphrases that conveyed similar meaning, resulting in comparable reward values. This suggests that simpler sentences may be less sensitive to prompt variation. The second input which focused on artificial intelligence transforming industries, showed a modest improvement after optimization: the "TL;DR:" prompt produced a slightly more concise and semantically aligned summary, increasing the reward from 0.312 to 0.376. Together, these findings indicate that while the degree of improvement varies depending on the input text, the RL agent consistently converged on "TL;DR:" as the most effective template and demonstrated the ability to enhance summarization quality across diverse examples.

6. Discussion

The results of this project demonstrate that even a lightweight reinforcement learning framework, operating on a small dataset and using a compact model such as FLAN-T5-base, can meaningfully improve prompt selection for summarization tasks. The improvements observed particularly the substantial gain in Input 3 highlight how sensitive LLMs are to instruction phrasing and how reinforcement learning can exploit this sensitivity to optimize model behavior. By using ROUGE-1 and BLEU as automated reward signals, the system eliminated the need for human feedback, enabling rapid experimentation in resource-constrained environments like Google Colab.

One important observation is that the degree of improvement varied across examples. While the spacecraft passage showed dramatic gains, the other two examples exhibited more modest or negligible changes. This variation reflects an inherent property of LLMs: the effect of prompt phrasing depends heavily on the structure, clarity, and lexical richness of the input text. Inputs that are already simple, straightforward, or highly compressed tend to leave little room for meaningful improvement. In contrast, passages with richer descriptive content benefit more from effective prompt framing, resulting in larger reward increases after optimization.

Another key insight is the consistency with which the Q-learning agent converged toward the same template ("TL;DR:") across different inputs. This suggests that certain prompt styles may be broadly effective for summarization, at least under lexical-overlap-based evaluation metrics. However, this convergence also highlights a limitation: ROUGE and BLEU tend to reward outputs that closely match the reference text, which may bias the agent toward templates that encourage copying rather than abstraction. In practical summarization systems, human evaluation or semantic-similarity metrics might be needed to ensure that the generated summaries are both accurate and meaningfully compressed.

Despite these limitations, the RLHF-Lite approach shows promising potential. It demonstrates that prompt optimization traditionally a subjective and manual process can be automated through simple reinforcement learning techniques. The results serve as a proof of concept that even without large-scale datasets, reward models, or advanced RL algorithms, meaningful gains can be achieved by systematically exploring and adapting prompts. This lightweight framework could be extended in future work by increasing the diversity of prompt templates, experimenting with alternative reward signals (e.g., BERTScore or semantic similarity), or applying the method to more complex generative tasks.

7. Future Work

Although this RLHF-Lite system demonstrates that reinforcement learning can guide prompt optimization for summarization, important extensions remain. One improvement is replacing lexical metrics like ROUGE with stronger semantic reward models, such as BERTScore [1] and BLEURT [2], which measure meaning

rather than word overlap. These reward models have become standard in modern summarization research and would likely provide more stable and meaningful signals for the agent.

Similarly, the Q-learning setup in this project can be extended using more advanced RL algorithms. Modern alignment pipelines use Proximal Policy Optimization (PPO) as described in InstructGPT [3], and newer approaches like GRPO used by DeepSeek-R1 enable "pure RL" training of language models. These methods allow multi-step prompt refinement rather than selecting from fixed templates, enabling more complex optimization behavior. Scaling the experiment to real summarization datasets such as XSUM or CNN/DailyMail would also allow the model to generalize beyond handcrafted samples.

Finally, future work could explore real human preference signals instead of purely automated metrics, aligning with lightweight RLHF methods from recent literature. With quantization tools (e.g., 4-bit QLoRA) and instruction-tuned LLMs, the same RLHF-lite framework could be extended to rewriting, question answering, and style transformation tasks. These enhancements would move this project closer to full RLHF and reflect the direction of current research in LLM alignment.

8. Conclusion

This project demonstrated the effectiveness of a lightweight reinforcement learning framework RLHF-Lite for optimizing prompt selection in Large Language Models. By combining a compact instruction-tuned model (FLAN-T5-base) with a simple Q-learning agent and automated evaluation metrics, the system successfully learned which prompt templates produced higher-quality summaries. The results, particularly the substantial improvement observed in the spacecraft input show that even without large datasets, human feedback, or complex reward models, reinforcement learning can meaningfully enhance LLM behavior in resource-constrained environments.

The experiment also highlighted important characteristics of prompt sensitivity in LLMs. Some passages benefited significantly from optimized prompting, while others showed only minor changes, revealing that the impact of reinforcement learning depends on the linguistic structure of the input text. Nonetheless, the

consistent convergence toward the "TL;DR:" template indicates that certain instructions may be broadly effective under ROUGE/BLEU-based evaluation. These findings confirm that prompt engineering is not only a critical aspect of LLM performance but also one that can be automated through systematic exploration.

Overall, this project provides a clear proof of concept for using reinforcement learning as a practical tool for prompt optimization. It opens pathways for future work such as exploring more diverse prompt sets, evaluating with semantic similarity metrics, scaling to larger datasets, or applying the RLHF-Lite framework to tasks beyond summarization, such as question answering or style transformation. The results reaffirm that even simple RL methods can offer valuable improvements and deepen our understanding of how LLMs respond to prompt-level guidance.

9. References

[1] T. Zhang *et al.*, "BERTScore: Evaluating Text Generation with BERT," *arXiv preprint arXiv:1904.09675*, 2020.

[2] T. Sellam, D. Das, and A. Parikh, "BLEURT: Learning Robust Metrics for Text Generation," *arXiv preprint arXiv:2004.04696*, 2020.

[3] L. Ouyang *et al.*, "Training language models to follow instructions with human feedback," *arXiv preprint arXiv:2203.02155*, 2022.

[4] Chung, Hyung Won, et al. "Scaling instruction-finetuned language models." Journal of Machine Learning Research 25.70 (2024): 1-53.

CSI5130-Artificial-Intelligence---project/Adaptive_Prompt_RLHF.ipynb at main · agnaantony26/CSI5130-Artificial-Intelligence---project