Bachelor of Science in Electrical and Electronic Engineering

EEE 400 (July 2023): Thesis

Enhanced Generative Question Answering Using Large Language Models and Reinforcement Learning from Human Feedback

Submitted by

Usama Ibna Alam
1806026

Sudipto Pramanik
1806172

Supervised by

Dr. Mohammad Ariful Haque
Professor

Department of Electrical and Electronic Engineering

Bangladesh University of Engineering and Technology

Dhaka, Bangladesh

June 2024

CANDIDATES' DECLARATION

This is to certify that the work presented in this thesis, titled, "Enhanced Generative Question Answering Using Large Language Models and Reinforcement Learning from Human Feedback", is the outcome of the investigation and research carried out by us under the supervision of Dr. Mohammad Ariful Haque.

It is also declared that neither this thesis nor any part thereof has been submitted anywhere else for the award of any degree, diploma or other qualifications.

Usama Ibna Alam
1806026

Sudipto Pramanik
1806172

CERTIFICATION

This thesis titled, **"Enhanced Generative Question Answering Using Large Language Models and Reinforcement Learning from Human Feedback"**, submitted by the group as mentioned below has been accepted as satisfactory in partial fulfillment of the requirements for EEE 400: Project/Thesis course, and as the requirements for the degree B.Sc. in Electrical and Electronic Engineering in June 2024.

Group Members:

Usama Ibna Alam

Sudipto Pramanik

Supervisor:

Dr. Mohammad Ariful Haque
Professor
Department of Electrical and Electronic Engineering
Bangladesh University of Engineering and Technology

ACKNOWLEDGEMENT

We would like to express our deepest gratitude to our supervisor Dr. Mohammad Ariful Haque for his guidance on this project showing us the path of conducting successful research and above all for always being there as our mentor. He shared his wisdom with us in analyzing subject matters and at the same time valued our thinking approach to synthesize those topics. His suggestions drove us towards better ways of thinking, his reviews enriched us in solving problems, and his support gave us strength at the time of our disappointment. We shall forever cherish the memories of working with him. We deeply thank our friends and families for always believing in us even at the moment when we were losing our confidence.

Dhaka
June 2024

Usama Ibna Alam

Sudipto Pramanik

Contents

List of Figures

List of Tables

ABSTRACT

This thesis investigates the application of Reinforcement Learning from Human Feedback (RLHF) to enhance the performance of generative question-answering systems using large language models (LLMs), specifically GPT-2 and LLaMA2-7B. Traditional LLMs, while powerful, often struggle to maintain consistency and relevance in their responses, particularly in complex generative tasks such as storytelling and letter writing. This research demonstrates how RLHF can be effectively applied to tune these models, aligning them more closely with human preferences and improving the quality of their output. Through a series of experiments, we present quantitative and qualitative analyses that highlight significant performance improvements in the generative capabilities of these models post-RLHF tuning. The findings of this study contribute to the broader understanding of how human feedback can be leveraged to refine the outputs of LLMs, paving the way for more nuanced and contextually appropriate generative text applications.

Keywords: Large Language Models (LLMs), GPT-2, LLaMA2-7B, Reinforcement Learning from Human Feedback (RLHF).

Chapter 1

Introduction

The rapid advancement of Artificial Intelligence (AI) has revolutionized various domains, with Large Language Models (LLMs) like GPT-2 and LLaMA-2 at the forefront of these innovations. These models have shown impressive capabilities in generating texts and answering questions with good accuracy. However, the pursuit of optimizing these models for more complex and contextually rich tasks remains a significant challenge. This thesis explores the integration of Reinforcement Learning from Human Feedback (RLHF) to enhance the performance of generative question answering systems, aiming to provide more accurate, relevant, and human-like responses.

1.1 Motivation

The motivation behind this research stems from the growing demand for reliable and sophisticated AI-driven question answering systems. Current models like GPT-2 and LLaMA-2 have demonstrated substantial potential, yet they struggle with consistency and contextual accuracy in complex queries, often generating responses that seem machine-like rather than human-like. The integration of Reinforcement Learning from Human Feedback (RLHF) aims to address these issues by refining the models to produce more natural and contextually relevant answers. Improving these models can lead to significant advancements in various fields such as education, where enhanced AI tutors can provide personalized learning experiences, customer service, where AI models can handle inquiries more accurately and reduce response times, healthcare, where reliable AI can assist professionals with quick access to accurate medical information, and legal assistance, where AI can support legal professionals by retrieving relevant case laws and drafting documents, thus saving time and reducing workload.

1.2 Challenges

The integration of RLHF with LLMs presents several challenges that must be addressed to achieve the desired improvements:

- **Data Diversity and Quality:** Ensuring the availability of diverse and high-quality datasets is crucial for training models effectively. The inclusion of varied topics and contexts helps in generalizing the models better.
- **Computational Resources:** Managing the substantial computational resources required for training and fine-tuning large language models is a significant challenge. Efficient use of these resources is essential to reduce overall costs.
- **Model Safety and Reliability:** Ensuring that the models generate safe and reliable responses, especially in critical applications like healthcare and legal assistance, is paramount. This includes preventing the propagation of incorrect or harmful information.
- **Human Feedback Integration:** Effectively integrating human feedback into the training process is challenging, requiring efficient mechanisms to collect, interpret, and incorporate feedback into the models without introducing noise or biases.
- **Scalability:** Scaling the RLHF approach to larger models and diverse datasets while maintaining performance and efficiency is a significant challenge. Ensuring that the approach works across different domains and applications requires careful design and implementation.

1.3 Objectives of the Thesis

The primary objectives of this thesis are:

- **Enhance Model Performance:** Improve the accuracy, relevance, and human-likeness of responses generated by GPT-2 and LLaMA2-7B through the integration of RLHF. This involves refining the models to handle more complex and contextually rich queries effectively.
- **Evaluate Effectiveness:** Compare the performance of GPT-2 and LLaMA2-7B before and after applying RLHF to determine the relative improvement. This comparison will help in understanding the strengths and weaknesses of each model and the impact of RLHF.

- **Ensure Safety and Reliability:** Develop models that are safe, reliable, and capable of generating high-quality responses consistently. This involves rigorous testing and validation to prevent the generation of harmful or incorrect outputs.

1.4 Contribution

This thesis makes several key contributions to the field of AI and generative question answering:

- **Integration of RLHF:** Demonstrates the integration of RLHF with GPT-2 and LLaMA2-7B, showing significant improvements in model performance. The study provides a detailed methodology for incorporating human feedback into the training process.
- **Performance Evaluation:** Provides a comprehensive evaluation of the models, highlighting the enhancements in accuracy, consistency, and reliability. The evaluation includes various parameters to assess the quality of the generated responses.
- **Comparative Analysis:** Offers a detailed comparison between GPT-2 and LLaMA2-7B, illustrating the relative improvements and strengths of each model. This analysis helps in understanding which model benefits more from RLHF and under what conditions.
- **Human-like Generation:** Enhances the ability of models to generate responses that are more natural, contextually relevant, and human-like by effectively integrating human feedback. This contribution focuses on reducing the machine-like nature of responses to improve user experience and applicability across various fields.

1.5 Thesis Outline

The thesis is structured as follows:

- **Chapter 1- Introduction:** This chapter outlines the motivation, challenges, objectives, and contributions of the thesis.
- **Chapter 2- Literature Review:** Reviews existing work related to LLMs, RLHF, and generative question answering. This chapter provides the background and context for the research, summarizing key developments and methodologies in the field.
- **Chapter 3- Methodology:** Describes the approach and techniques used for integrating RLHF with LLMs. This includes details on language model training, and the implementation of RLHF.

- **Chapter 4- Experimental Setup:** This chapter outlines the descriptions of the models used, overall workflow, design considerations and evaluation framework.

- **Chapter 5: Results and Discussion:** This chapter details the results and presents the findings of the study, including various metrics and visual comparisons of the models' performance. It also discusses on broader implications and limitations of the study.

- **Chapter 6: Conclusion:** Provides a precise summary of the study and suggests for possible future works.

This thesis aims to advance the field of generative question answering by enhancing the capabilities of LLMs through RLHF, contributing to more effective and reliable AI systems. The research demonstrates the potential of RLHF in improving the performance of GPT-2 and LLaMA2-7B, offering valuable insights for future developments in AI.

Chapter 2

Literature Review

2.1 Evolution of Large Language Models

The field of natural language processing (NLP) has witnessed significant advancements with the evolution of large language models (LLMs). Early models like BERT and GPT-1 played crucial roles in this evolution. BERT, introduced by Devlin et al. [1], utilized a bidirectional training approach to capture context from both directions, significantly improving performance on various NLP tasks such as named entity recognition, question answering, and language inference [2]. This bidirectional context capture marked a significant departure from previous models, which typically processed text in a single direction. GPT-1, introduced by Radford et al. [3], demonstrated the power of unsupervised pre-training followed by fine-tuning on specific tasks. This model set the stage for autoregressive language models that generate coherent and contextually appropriate text by predicting the next word in a sequence. This approach was further refined and scaled up in subsequent models like GPT-2 and GPT-3, which have become benchmarks in the field of generative pre-training [4]. These models showcased the potential of using large-scale datasets and transformer architectures to significantly improve the quality and coherence of generated text [5].

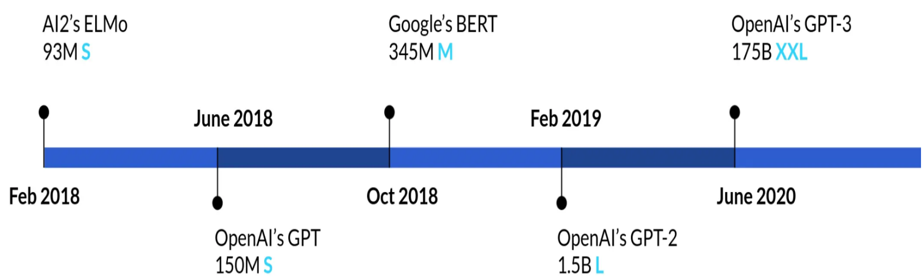


Figure 2.1: Evolution of LLMs

2.2 Capabilities of Large Language Models

LLMs have shown remarkable versatility across numerous applications. BERT and its derivatives have been effectively utilized for tasks such as text classification, sentiment analysis, and named entity recognition, demonstrating their broad utility in NLP [6]. The introduction of GPT-3 expanded these applications significantly, excelling in creative writing, code generation, and complex problem-solving tasks [7]. Furthermore, LLMs have found applications in specialized fields such as healthcare and legal document analysis, where they assist in diagnosis, treatment planning, and legal research [8,9]. In healthcare, LLMs have enabled the development of systems that can provide preliminary diagnoses, recommend treatments, and even assist in medical research by generating hypotheses based on existing data [10]. Similarly, in the legal field, LLMs help in drafting documents, conducting legal research, and interpreting complex legal language, thus saving significant time and reducing workload for legal professionals [11].

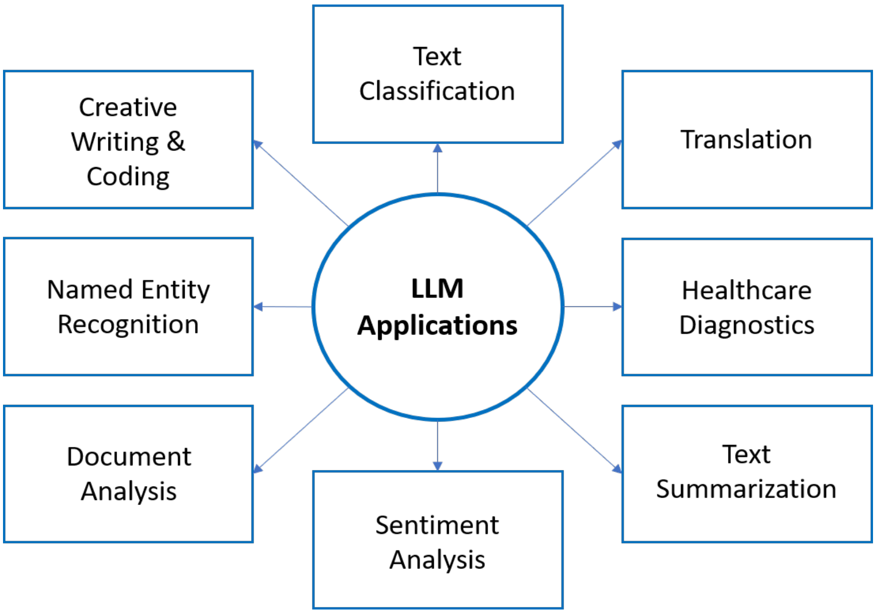


Figure 2.2: LLM Applications

2.3 LLMs in Question Answering

The application of LLMs to question answering (QA) has been a focal point of research, transforming traditional QA systems. Early QA systems relied heavily on rule-based approaches or smaller models that struggled with deep contextual understanding. The advent of models like GPT-2 marked a significant leap in QA capabilities. GPT-2's ability to generate coherent and contextually relevant responses improved QA accuracy [12, 13]. GPT-2's architecture, which builds on the autoregressive model of GPT-1, enabled it to handle a wide range of tasks by generating text that is coherent and contextually appropriate. This versatility made GPT-2 particularly effective for QA systems, where the ability to understand and generate contextually relevant responses is crucial [14]. Recent advancements have seen the integration of LLMs with other models to improve QA performance further. For instance, RAG (Retrieval-Augmented Generation) combines the strengths of retrieval-based and generative models, providing a robust framework for generating accurate and contextually relevant answers [15]. Similarly, the integration of knowledge graphs with LLMs has shown promise in improving the factual accuracy of generated responses [16].

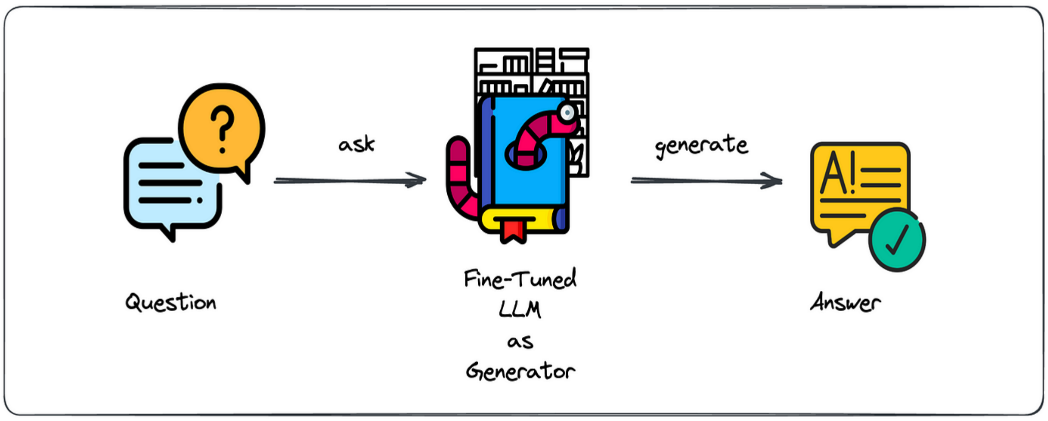


Figure 2.3: A Fine-Tuned LLM in QA System

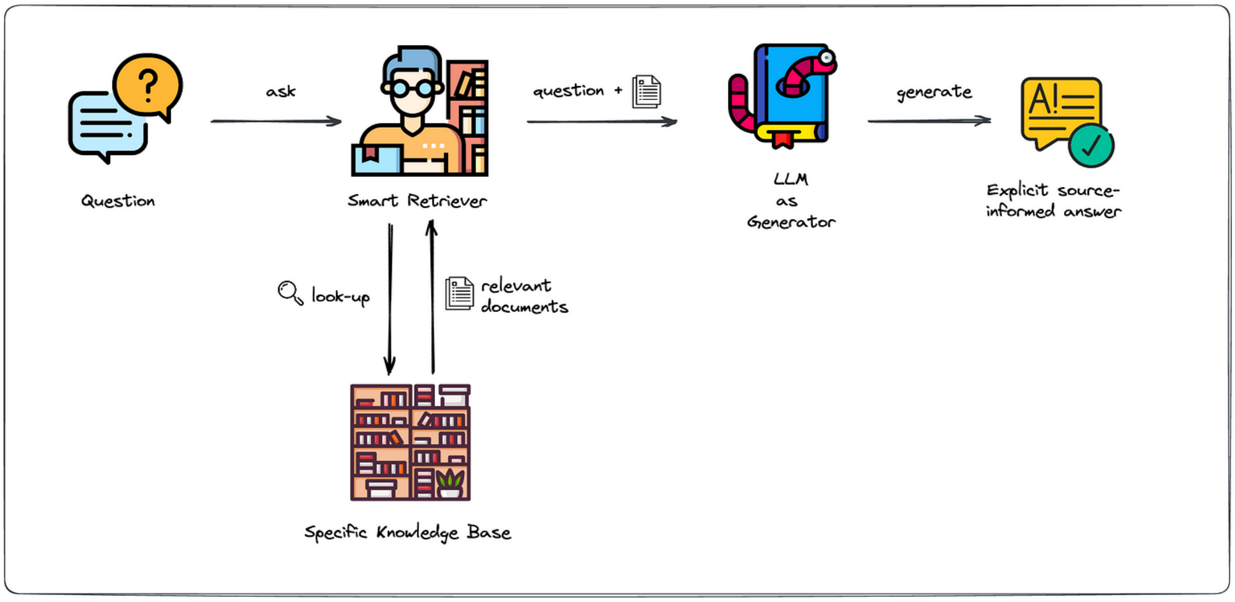


Figure 2.4: An Advanced QA System Using LLMs

2.4 Introduction to RLHF

Reinforcement Learning (RL) involves training models to make a sequence of decisions by performing actions in an environment to maximize cumulative rewards. Sutton and Barto's foundational work on RL laid the groundwork for its integration with LLMs [9]. The concept of Reinforcement Learning from Human Feedback (RLHF) has gained traction, where human feedback is utilized to guide the learning process, aligning model outputs more closely with human preferences [10]. RLHF is particularly useful in scenarios where predefined rules or objectives are difficult to specify. By incorporating human feedback, RLHF allows models to learn complex, nuanced behaviors that align with human expectations. This approach has been used to fine-tune models for tasks such as content generation, dialogue systems, and recommendation systems, where the quality and relevance of the output are critical [17].

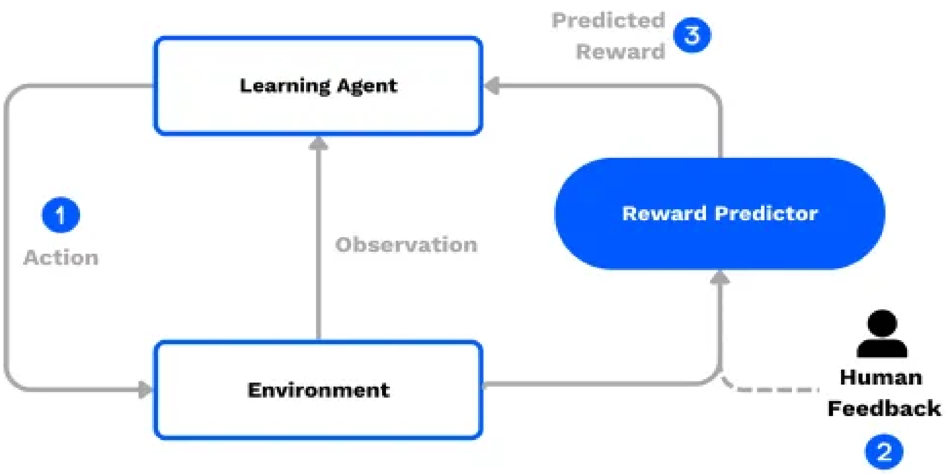


Figure 2.5: Reinforcement Learning from Human Feedback (RLHF)

RL has found applications in diverse fields, ranging from robotics to game playing and natural language processing. In NLP, RL has been employed to optimize dialogue systems, enhance machine translation, and improve text generation through fine-tuning based on feedback or reward signals [18]. The integration of RL with LLMs aims to address challenges such as generating contextually accurate and human-like responses, thereby improving user experience [19]. One notable application of RL in NLP is in optimizing dialogue systems. By using RL to adjust the responses based on user feedback, these systems can provide more accurate and relevant answers, improving user satisfaction. Similarly, in machine translation, RL helps fine-tune models to produce more natural and contextually appropriate translations by rewarding outputs that align closely with human expectations [12].

2.5 Potential Use of RLHF in Question Answering

Applying RL to answer generation involves using reward models to evaluate the quality of generated answers and fine-tuning LLMs based on these evaluations. Techniques like Proximal Policy Optimization (PPO) are commonly used to optimize model parameters effectively. Studies have shown that RLHF can significantly enhance the performance of LLMs in generating high-quality, contextually relevant answers, making it a promising approach for QA systems [18,20]. RLHF leverages human feedback to refine the responses generated by LLMs, ensuring they are not only accurate but also contextually appropriate and human-like. This approach has proven effective in various applications, including customer service bots and automated tutoring systems, where the quality of the generated answers is paramount [10].

Previous works have explored the integration of RLHF to improve the performance of LLMs. For example, Christiano et al. [10] demonstrated how human feedback could be used to train models to align with complex human values, significantly enhancing the quality of generated text. Similarly, Ouyang et al. [4] showed that RLHF could be effectively employed to train language models to follow instructions more accurately, resulting in more contextually appropriate and human-like responses. Another notable study by Ziegler et al. [20] focused on fine-tuning LLMs using RLHF for tasks requiring high levels of creativity and contextual understanding, such as story generation. Their work highlighted the potential of RLHF in significantly improving the human-likeness and contextual relevance of generated text, making it a promising approach for enhancing QA systems.

Despite significant advancements, several challenges remain in the field of LLMs and their application to QA systems. Firstly, while LLMs like GPT-2 have demonstrated impressive capabilities, they still struggle with consistency and contextual accuracy, particularly in complex queries. Current models often generate responses that, while coherent, may lack the depth of understanding required for nuanced and contextually rich answers [13, 14]. Moreover, traditional fine-tuning methods, although effective to some extent, do not fully address the limitations in aligning model outputs with human expectations. This gap is particularly evident in applications where the quality of the generated response must closely match human-like understanding and reasoning [18].

This thesis addresses these challenges by integrating RLHF with LLMs to enhance their performance in QA tasks. The key contributions of this work include demonstrating the effectiveness of RLHF in improving the quality of answers generated by LLMs, comparing the performance of different LLMs, such as GPT-2 and LLaMA2-7B, to identify the most effective models for QA tasks, and providing insights into the practical applications of RLHF-enhanced LLMs in various domains.

Chapter 3

Methodology

3.1 Generation Using LLM

This section outlines the process of fine-tuning Large Language Models (LLMs) for question answering, emphasizing the adjustment of pre-trained models to accurately respond to queries with contextually relevant answers.

3.1.1 What are Large Language Models (LLMs)?

A large language model (LLM) is a type of artificial intelligence (AI) algorithm that uses deep learning techniques and massively large data sets to understand, summarize, generate and predict new content. The term generative AI also is closely connected with LLMs, which are, in fact, a type of generative AI that has been specifically architected to help generate text-based content.

Over millennia, humans developed spoken languages to communicate. Language is at the core of all forms of human and technological communications; it provides the words, semantics and grammar needed to convey ideas and concepts. In the AI world, a language model serves a similar purpose, providing a basis to communicate and generate new concepts.

The first AI language models trace their roots to the earliest days of AI. The Eliza language model debuted in 1966 at MIT and is one of the earliest examples of an AI language model. All language models are first trained on a set of data, and then they make use of various techniques to infer relationships and then generate new content based on the trained data. Language models are commonly used in natural language processing (NLP) applications where a user inputs a query in natural language to generate a result.

An LLM is the evolution of the language model concept in AI that dramatically expands the data used for training and inference. In turn, it provides a massive increase in the capabilities of the AI model. While there isn't a universally accepted figure for how large the data set for training needs to be, an LLM typically has at least one billion or more parameters. Parameters are a machine learning term for the variables present in the model on which it was trained that can be used to infer new content.

Modern LLMs emerged in 2017 and use transformer models, which are neural networks commonly referred to as transformers. With a large number of parameters and the transformer model, LLMs are able to understand and generate accurate responses rapidly, which makes the AI technology broadly applicable across many different domains. [21] [22]

3.1.2 How do Large Language Models work?

LLMs take a complex approach that involves multiple components.

At the foundational layer, an LLM needs to be trained on a large volume – sometimes referred to as a corpus – of data that is typically petabytes in size. The training can take multiple steps, usually starting with an unsupervised learning approach. In that approach, the model is trained on unstructured data and unlabeled data. The benefit of training on unlabeled data is that there is often vastly more data available. At this stage, the model begins to derive relationships between different words and concepts.

The next step for some LLMs is training and fine-tuning with a form of self-supervised learning. Here, some data labeling has occurred, assisting the model to more accurately identify different concepts.

Next, the LLM undertakes deep learning as it goes through the transformer neural network process. The transformer model architecture enables the LLM to understand and recognize the relationships and connections between words and concepts using a self-attention mechanism. That mechanism is able to assign a score, commonly referred to as a weight, to a given item (called a token) in order to determine the relationship.

Once an LLM has been trained, a base exists on which the AI can be used for practical purposes. By querying the LLM with a prompt, the AI model inference can generate a response, which could be an answer to a question, newly generated text, summarized text or a sentiment analysis report. [21] [22]

3.1.3 What is Fine-tuning, and Why is it Important?

Fine-tuning is the process of taking a pre-trained model and further training it on a domain-specific dataset.

Most LLM models today have a very good global performance but fail in specific task-oriented problems. The fine-tuning process offers considerable advantages, including lowered computation expenses and the ability to leverage cutting-edge models without the necessity of building one from the ground up.

Transformers grant access to an extensive collection of pre-trained models suited for various tasks. Fine-tuning these models is a crucial step for improving the model's ability to perform specific tasks, such as sentiment analysis, question answering, or document summarization, with higher accuracy. [23] [24]

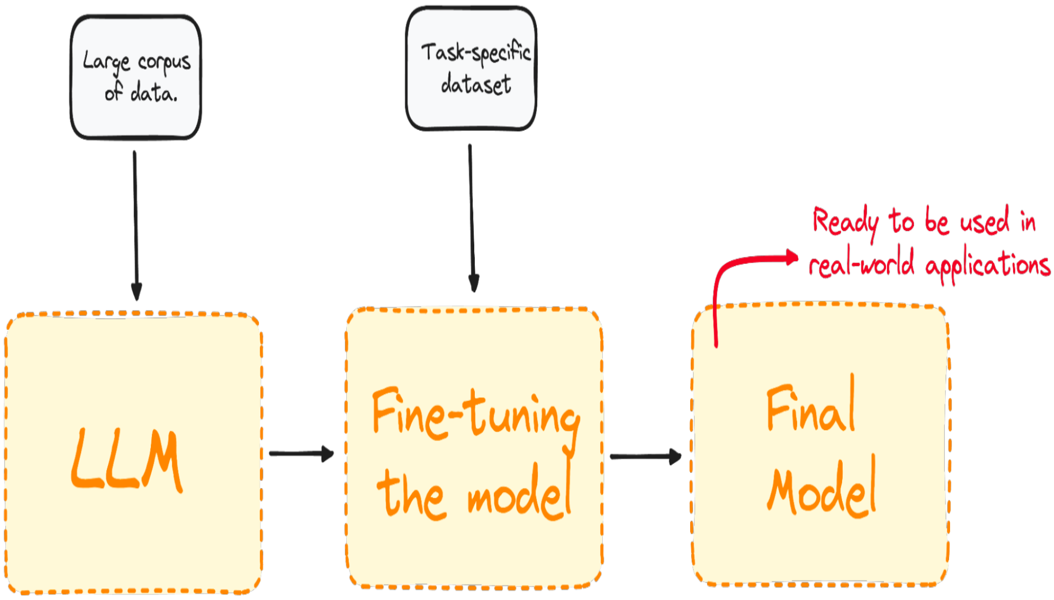


Figure 3.1: Visualizing the Fine-Tuning Process

3.1.4 Steps for Fine-tuning a LLM

(i) **Step 1- Choosing a Pre-trained Model and a Dataset to Fine-tune the Model:** We always need to have a pre-trained model in mind appropriate with the target task. A proper dataset is also required.

(ii) **Step 2- Load the Data to Use:** Now that we have our model, we need some good-quality data to work with, and this is precisely where the datasets library kicks in to load the dataset in the working environment.

(iii) **Step 3- Tokenizer:** Now that we already have our dataset, we need a tokenizer to prepare it to be parsed by our model.As LLMs work with tokens, we need to load a pre-trained Tokenizer and tokenize our dataset so it can be used for fine-tuning.

(iv) **Step 4- Train-Test Split:** To improve processing requirements, we create a smaller subset of the full dataset to fine-tune our model. The training set will be used to fine-tune the model, while the testing set will be used to evaluate it.

(v) **Step 5- Initialization of the Base Model:** Loading the base model and initializing it properly.

(vi) **Step 6- Evaluate Method:** Transformers provides a Trainer class optimized for training. However, this method does not include how to evaluate the model. This is why, before starting the training, we need to pass Trainer a function to evaluate our model performance.

(vii) **Step 7- Fine-tune Using the Trainer Method:** Our final step is to set up the training arguments and start the training process. The Transformers library contains the Trainer class, which supports a wide range of training options and features such as logging, gradient accumulation, and mixed precision. We first define the training arguments together with the evaluation strategy. Once everything is defined, we can easily train the model simply using the train() command. After training, evaluate the model's performance on the validation or test set. Again, the trainer class already contains an evaluate method that takes care of this. [23]

The above process can be applied to fine-tune a LLM like GPT-2 or LLAMA2-7B to improve the model's ability in context based question answering.

3.1.5 Modern Techniques in Fine-tuning

PEFT

Parameter-Efficient Fine-Tuning (PEFT) is a technique designed to optimize the fine-tuning of large pre-trained models by adjusting only a small subset of parameters. This approach substantially reduces computational costs and memory usage while maintaining high performance, making it particularly useful for adapting models to specific downstream tasks.

One key method in PEFT involves the use of adapter layers. These small neural network modules are inserted between the layers of a pre-trained model and are the only components trained during the fine-tuning process. This allows the core parameters of the pre-trained model to remain unchanged, facilitating efficient task-specific adjustments without substantial computational overhead. Adapter-based fine-tuning is recognized for its flexibility and modularity, though it may require careful hyperparameter tuning to achieve optimal performance [25, 26].

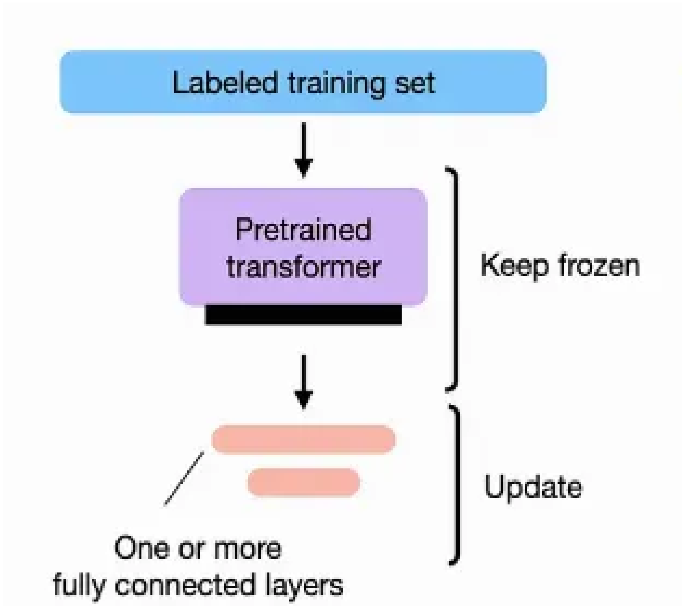


Figure 3.2: Concept of PEFT

Bias tuning, also known as BitFit, is another significant PEFT method that focuses on fine-tuning only the bias terms in the model's layers. This drastically reduces the number of parameters involved, making the process highly efficient. BitFit achieves substantial parameter reduction while maintaining performance levels close to those of fully fine-tuned models [27, 28].

The benefits of PEFT are particularly evident in scenarios where computational resources are limited. For instance, fine-tuning large language models on consumer-grade hardware, such as Nvidia GPUs with limited VRAM, becomes feasible with PEFT methods. This is crucial for applications requiring rapid adaptation to new domains with limited data and for scalable AI deployments where maintaining computational efficiency is essential [25, 28].

LoRA

Low-Rank Adaptation (LoRA) is a highly efficient technique used in Parameter-Efficient Fine-Tuning (PEFT) to adapt large pre-trained models by reducing the number of trainable parameters. LoRA achieves this by decomposing weight updates into low-rank matrices, thereby minimizing computational and memory requirements.

The main idea of LoRA is to approximate a pre-trained model's weight matrix W using the product of two smaller matrices A and B, where $W \approx AB$. During fine-tuning, only these low-rank matrices A and B are updated, significantly reducing the number of parameters involved. This method allows efficient fine-tuning even on hardware with limited computational resources, making it feasible to adapt large models to specific tasks without extensive infrastructure [29].

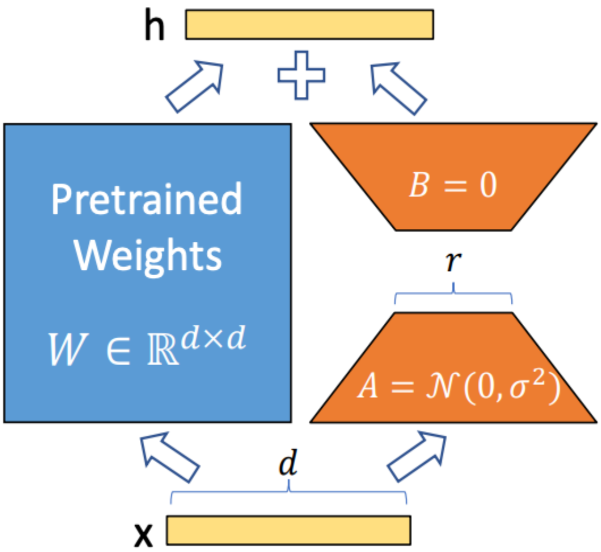


Figure 3.3: Illustration of LoRA Process

LoRA is particularly useful for fine-tuning large language models (LLMs) like GPT-3 on consumer-grade hardware, allowing for high performance with reduced computational load. This makes it an attractive option for various applications where computational efficiency is crucial [30].

3.2 Enhancing LLM Performance through RLHF

This section outlines how Reinforcement Learning from Human Feedback (RLHF) integrates human feedback into Large Language Model (LLM) training, marking a significant advancement in developing models capable of generating text that deeply aligns with complex human values and preferences.

3.2.1 What is RLHF?

Reinforcement learning from human feedback (RLHF) is a machine learning (ML) technique that uses human feedback to optimize ML models to self-learn more efficiently. Reinforcement learning (RL) techniques train software to make decisions that maximize rewards, making their outcomes more accurate. RLHF incorporates human feedback in the rewards function, so the ML model can perform tasks more aligned with human goals, wants, and needs. RLHF is used throughout generative artificial intelligence (generative AI) applications, including in large language models (LLM). [31] [32]

3.2.2 How does RLHF work?

RLHF is performed in four stages before the model is considered ready.

(i) **Data Collection:** Before performing ML tasks with the language model, a set of human-generated prompts and responses are created for the training data. This set is used later in the model's training process.

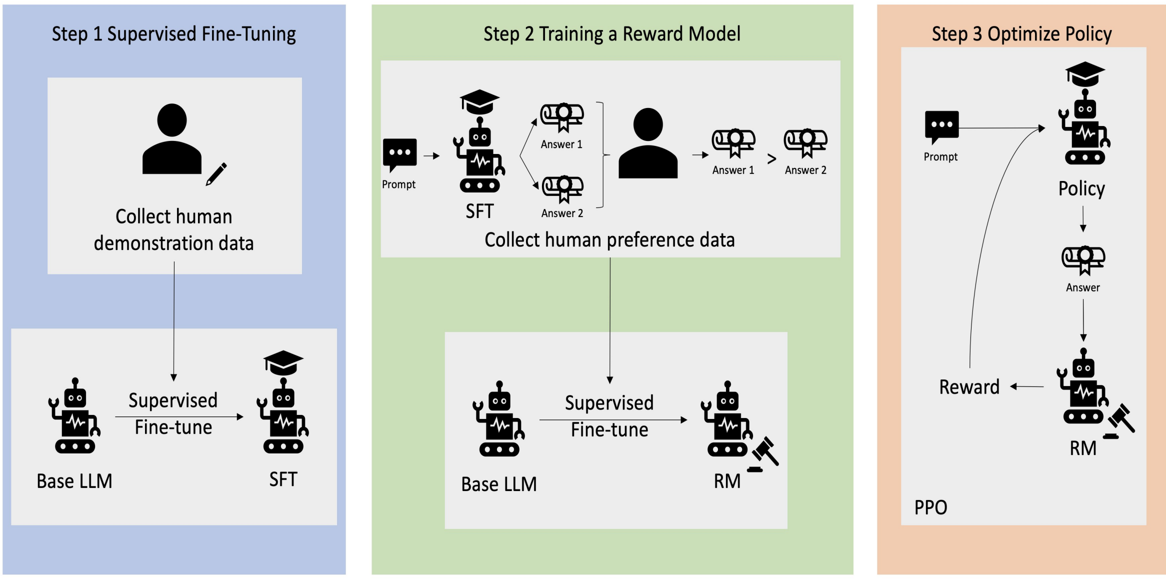


Figure 3.4: overview of the RLHF learning process

(ii) **Supervised Fine-tuning of a Language Model:** A commercial pretrained model can be used as the base model for RLHF. The model is fine-tuned to the company's internal knowledge base by using techniques such as retrieval-augmented generation (RAG). When the model is fine-tuned, its response to the predetermined prompts is compared with the human responses collected in the previous step. Mathematical techniques can calculate the degree of similarity between the two.

For example, the machine-generated responses can be assigned a score between 0 and 1, with 1 being the most accurate and 0 being the least accurate. With these scores, the model now has a policy that is designed to form responses that score closer to human responses. This policy forms the basis of all future decision-making for the model.

(iii) **Building a Separate Reward Model:** The core of RLHF is training a separate AI reward model based on human feedback, and then using this model as a reward function to optimize policy through RL. Given a set of multiple responses from the model answering the same prompt, humans can indicate their preference regarding the quality of each response. These response-rating preferences should be used to build the reward model that automatically estimates how high a human would score any given prompt response.

(iv) **Optimize the Language Model with the Reward-based Model:** The language model then uses the reward model to automatically refine its policy before responding to prompts. Using the reward model, the language model internally evaluates a series of responses and then chooses the response that is most likely to result in the greatest reward. This means that it meets human preferences in a more optimized manner. [31] [33]

Chapter 4

Experimental Setup

4.1 Model Description

4.1.1 GPT-2

GPT-2, which stands for 'Generative Pre-trained Transformer 2' is a state-of-the-art language model developed by OpenAI. It represents a significant advancement in natural language processing (NLP) and generative text models. Here's a detailed description of GPT-2, its architecture, capabilities, and applications:

Overview

GPT-2 is an autoregressive language model based on the Transformer architecture, which was first introduced by [34]. It is the successor to the original GPT model and is designed to generate coherent and contextually relevant text by predicting the next word in a sequence, given the preceding words.

Architecture

i. Transformer Architecture: GPT-2 employs the Transformer architecture, which relies on self-attention mechanisms to process and generate text. This architecture allows GPT-2 to capture long-range dependencies and context within the text effectively.

The architecture starts with input embeddings combined with positional encodings to retain word order information. The core of the Transformer, multi-head attention, allows the model to look at the information from multiple representation spaces at different positions concurrently. This data then passes through feed-forward neural networks individually for each position, integrated with normalization and skip connections to stabilize learning. The architecture concludes with linear layers and a softmax function to output probabilities, making it highly effective for a range of tasks requiring deep language understanding, such as translation and content generation. [35]

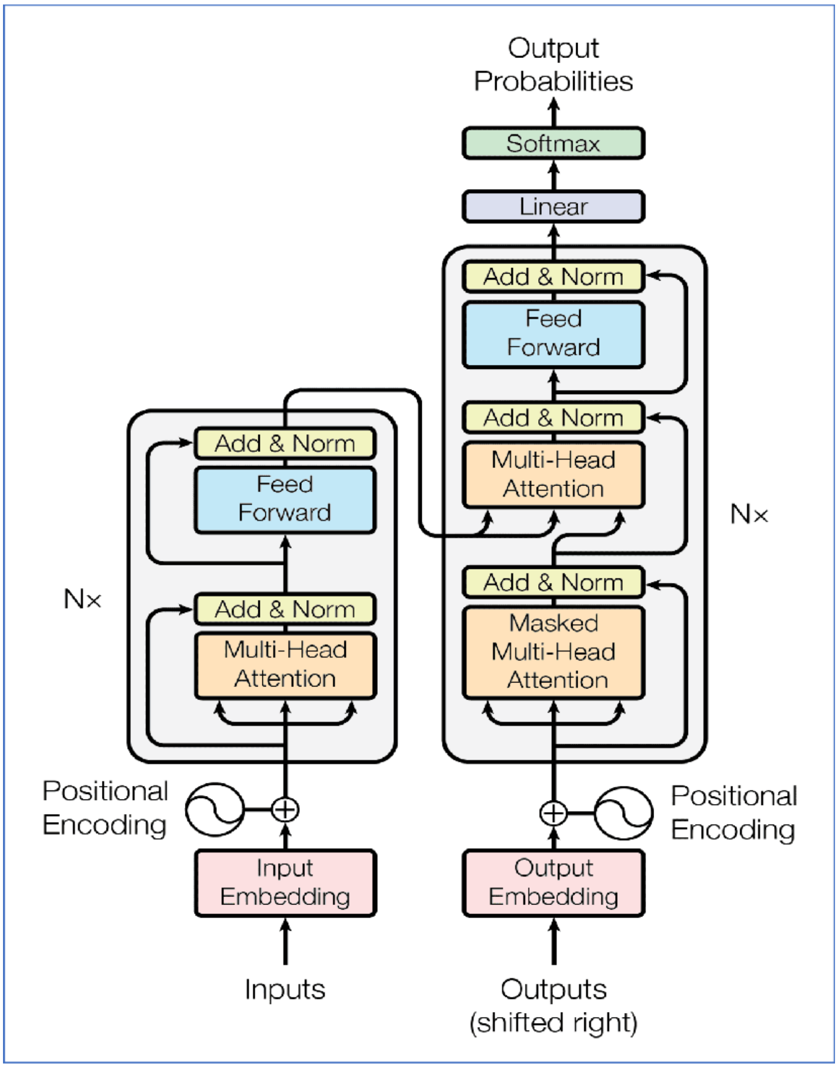


Figure 4.1: Transformer Architecture

ii. Layers and Parameters: GPT-2 comes in various sizes, with the largest model (known as GPT-2 XL) consisting of 1.5 billion parameters and 48 layers. Smaller versions include models with 117 million, 345 million, and 762 million parameters. In our thesis, we have utilized GPT-2 Large which has 762 million parameters. The large number of parameters enables GPT-2 to generate high-quality, coherent text. [36]

iii. Training Data: GPT-2 was trained on a diverse dataset called WebText, which comprises text from 8 million web pages. This dataset was curated to include high-quality content across various domains, ensuring that GPT-2 can generate text on a wide range of topics. [36]

Key Features

i. Autoregressive Model: GPT-2 generates text by predicting one word at a time, based on the words that precede it. This autoregressive nature allows for the generation of coherent and contextually relevant text. [37]

ii. Unsupervised Learning: GPT-2 is trained using unsupervised learning, meaning it learns to understand and generate text without explicit human annotations. This training method allows GPT-2 to leverage vast amounts of unlabeled text data.

iii. Few-Shot Learning: One of the standout features of GPT-2 is its ability to perform few-shot learning. Given a small number of examples, GPT-2 can adapt to new tasks and generate relevant responses. This capability makes it highly versatile for various applications.

Capabilities

i. Text Generation: GPT-2 excels at generating appropriate text. It can continue a given prompt, generate creative writing, and produce content in various styles and tones.

ii. Text Completion: Given a partial sentence or paragraph, GPT-2 can predict and complete the text, making it useful for drafting emails, articles, and other written content.

iii. Translation and Summarization: While not explicitly trained for these tasks, GPT-2 can perform translation and summarization to a certain extent by leveraging its language generation capabilities.

iv. Conversational Agents: GPT-2 can be used to build chatbots and conversational agents that provide human-like responses in a dialogue, enhancing user interaction experiences.

Applications

i. Content Creation GPT-2 is used for generating articles, blog posts, and creative writing. It can assist writers by providing inspiration and drafting text.

ii. Customer Support GPT-2 powers chatbots and virtual assistants in customer support applications, providing quick and coherent responses to user queries.

iii. Education and Tutoring GPT-2 can generate educational content, answer questions, and provide explanations on a wide range of topics, supporting learning and tutoring applications.

iv. Entertainment GPT-2 is employed in generating dialogues for video games, interactive storytelling, and other entertainment-related content.

Example of Loading GPT-2 and Generating Text

To use GPT-2 for text generation, one can perform following steps using the Hugging Face Transformers library:

```python
# Import necessary libraries
from transformers import AutoModelForCausalLM, AutoTokenizer

# Load pre-trained GPT-2
model = AutoModelForCausalLM.from_pretrained("gpt2")
tokenizer = tokenizer = AutoTokenizer.from_pretrained("gpt2")

# Tokenize input text
input_text = "Once upon a time in a distant galaxy,"
input_ids = tokenizer.encode(input_text, return_tensors='pt')

# Generate text
output = model.generate(input_ids, max_length=100, num_return_sequences=1,
    no_repeat_ngram_size=2)

# Decode and print the generated text
generated_text = tokenizer.decode(output[0], skip_special_tokens=True)
print(generated_text)
```

Listing 4.1: Loading GPT-2 and Generating Text

In this example, the `AutoModelForCausalLM` and `AutoTokenizer` classes from the Hugging Face Transformers library are used to load the GPT-2 model and tokenizer. The input text is tokenized and passed to the model for text generation. The `generate` method is used to produce a sequence of text based on the input, and the `decode` method converts the tokenized output back to readable text. [38]

4.1.2 LLaMA-2

LLaMA-2 (Large Language Model Meta AI 2) is an advanced language model developed by Meta, designed to generate human-like text with high coherence and contextual relevance. This model builds upon the foundational LLaMA model, enhancing its capabilities and performance in various natural language processing (NLP) tasks.

Architecture

LLaMA-2 utilizes a transformer-based architecture. Transformer architecture has been described in subsection 4.1.1 and shown in Figure 4.1 previously. It leverages self-attention mechanisms to process and generate text, enabling the model to capture long-range dependencies and maintain context effectively.

The model is available in several sizes comprising different number of parameters. In our thesis, we have utilized LLaMA2-7B which has seven billion parameters. These parameters empower LLaMA-2 to generate high-quality, coherent text across a wide range of applications. The model operates in an autoregressive manner, generating text one token at a time based on the preceding context. [39]

LLaMA-2 was trained on a diverse and extensive dataset curated from a variety of sources to ensure broad coverage of language use. This training approach enables LLaMA-2 to perform well across different domains, from casual conversation to technical writing. The training process involved significant computational resources and sophisticated optimization techniques to fine-tune the model's performance. [40]

Key Features and Capabilities

LLaMA-2 excels at generating coherent and contextually appropriate text. It can continue a given prompt, generate creative writing, and produce content in various styles and tones. The model's large parameter size allows it to understand and generate nuanced responses, making it suitable for tasks such as:

- **Long-Form Content Generation**: Generating extensive articles, essays, and reports that require maintaining context over long text spans [41].
- **Creative Writing**: Crafting stories, poems, and other creative content with a natural flow and engaging narrative.
- **Technical Documentation**: Assisting in the creation of detailed and accurate technical documents, manuals, and guides.

- **Interactive Storytelling**: Enhancing interactive storytelling applications by generating dynamic and engaging narratives based on user inputs [42].

LLaMA-2 also supports few-shot and zero-shot learning, enabling it to adapt to new tasks with minimal examples. This versatility makes it highly valuable for applications requiring flexibility and contextual understanding.

Applications

LLaMA-2 is used in various fields, including:

- **Content Creation**: Assisting in writing articles, blog posts, and other creative content.
- **Customer Support**: Powering chatbots and virtual assistants to improve customer interactions.
- **Education**: Generating educational content and providing tutoring assistance.
- **Entertainment**: Creating dialogues for video games and interactive storytelling.
- **Healthcare**: Assisting in drafting patient reports, summarizing medical records, and providing support in medical research.

Example of Loading LLaMA-2 and Generating Text

To use LLaMA-2 for text generation, you can follow these steps using the Hugging Face Transformers library:

```python
# Import necessary libraries
from transformers import LlamaTokenizer, LlamaForCausalLM

# Load pre-trained LLaMA-2 model and tokenizer
model_name = 'meta-llama/LLaMA-2'
model = LlamaForCausalLM.from_pretrained(model_name)
tokenizer = LlamaTokenizer.from_pretrained(model_name)

# Tokenize input text
input_text = "Once upon a time in a distant galaxy,"
input_ids = tokenizer.encode(input_text, return_tensors='pt')

# Generate text
output = model.generate(input_ids, max_length=100, num_return_sequences=1,
    no_repeat_ngram_size=2)

# Decode and print the generated text
generated_text = tokenizer.decode(output[0], skip_special_tokens=True)
print(generated_text)
```

Listing 4.2: Example of Loading LLaMA-2 and Generating Text

4.1.3 RoBERTa Base

RoBERTa (Robustly Optimized BERT Pretraining Approach) is a NLP model developed by Facebook AI, designed to improve upon the original BERT model by optimizing the pretraining process. This model is widely used in various natural language understanding tasks due to its robust performance.

Overview

RoBERTa is based on the BERT (Bidirectional Encoder Representations from Transformers) model but incorporates several key improvements. These enhancements include training the model longer, on larger batches, over more data, and removing the next sentence prediction objective. RoBERTa utilizes the masked language modeling (MLM) objective, where a certain percentage of the input tokens are masked, and the model is trained to predict these masked tokens based on the surrounding context. [43]

Architecture

The architecture of RoBERTa is fundamentally similar to BERT, consisting of a multi-layer Transformer encoder. It employs self-attention mechanisms to understand the context of words in a sentence bidirectionally. Here are some details:

i. Transformer Architecture: RoBERTa's architecture is based on the Transformer model, specifically focusing on improving BERT's training regimen. Transformer architecture has been described in subsection 4.1.1 and shown in Figure 4.1 previously.

ii. Layers and Parameters: RoBERTa Base model consists of 12 layers (transformer blocks), 768 hidden units per layer, and 12 attention heads, totaling approximately 125 million parameters. [44]

iii. Training Data: The model is pretrained on a large corpus of English data, including the BookCorpus, English Wikipedia, CC-News, OpenWebText, and Stories datasets. This comprehensive dataset covers diverse topics, enabling RoBERTa to generalize well across different domains. [45]

Key Features

i. Masked Language Modeling (MLM): RoBERTa uses MLM, where 15% of the input tokens are randomly masked, and the model predicts these masked tokens. This approach helps the model learn bidirectional representations, capturing context from both the left and right sides of a token. [46]

ii. Dynamic Masking: Unlike BERT, which uses a static mask during pretraining, RoBERTa employs dynamic masking, changing the mask pattern at each training epoch. This leads to more robust learning. [47]

iii. Increased Training Data and Steps: RoBERTa is trained on significantly more data (160GB) and for more steps than BERT, which enhances its performance on downstream tasks.

Capabilities

i. Text Classification: RoBERTa excels at text classification tasks, such as spam detection and topic categorization, by leveraging its deep understanding of language nuances.

ii. Named Entity Recognition (NER): The model can identify and classify entities (like names of people, organizations, locations) within a text.

iii. Text Summarization: While not specifically designed for text generation tasks, RoBERTa can be fine-tuned to provide concise summaries of long documents by understanding the main points.

iv. Sentiment Analysis: RoBERTa's nuanced understanding of language makes it highly effective for sentiment analysis tasks, allowing it to accurately interpret and categorize the emotional tone of text.

Applications

i. Content Moderation: Helps in identifying and filtering inappropriate or harmful content on social media platforms.

ii. Sentiment Analysis: Used in social media monitoring, customer feedback analysis, and market research to gauge public opinion and sentiment.

iii. Text Analysis: Assists in analyzing and categorizing documents, aiding in research and sensitive decision support systems.

4.2 Design

The overall design for our work is as follows:

1. Collection of Data:

Gather a comprehensive dataset that includes relevant question-answer pairs. Augment the dataset with additional annotations indicating the quality of the answers, such as “Good” and “Bad” labels.

2. Fine-tuning a Language Model:

Select a suitable Large Language Model (LLM) pre-trained on a large corpus of text data to establish a robust foundation for further fine-tuning. Fine-tune this model on custom dataset for better performance.

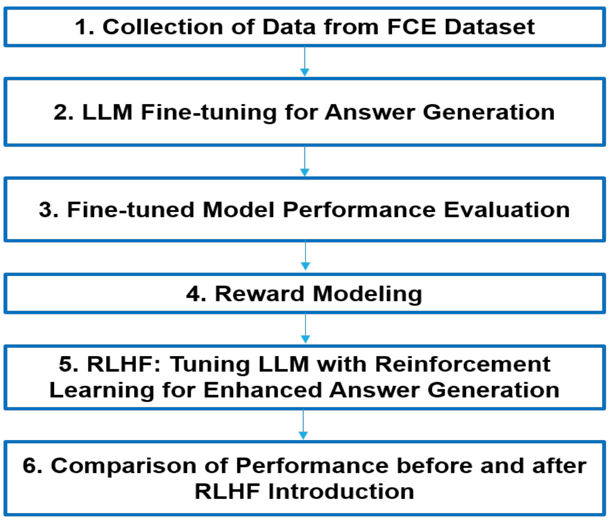


Figure 4.2: Overall Workflow

3. Fine-tuned Model Performance Evaluation:

Assess the performance of the fine-tuned model on a set of unique questions. This involves evaluating accuracy, relevance, fluency, and human-likeness of generated answers.

4. Train Reward Model:

Use a model appropriate to develop as a reward model. Fine-tune this model on the dataset to predict the quality labels and generate scalar rewards that reflect the quality of the answers.

5. RLHF; Fine-tune Language Model with Reward Model and Apply PPO:

Utilize the reward model’s scalar rewards to fine-tune the LLMs. Implement Proximal Policy

Optimization (PPO) to continuously optimize the LLMs. The PPO algorithm iteratively updates the models based on the reward signals, ensuring the models align more closely with human preferences and improve their performance in generating high-quality answers.

6. Evaluate the Performance:

Assess the performance on a set of unique questions. Compare reward values before and after the introduction of RLHF to evaluate improvements and ensure the models generate contextually relevant and accurate responses.

4.3 Design Considerations

In implementing Reinforcement Learning from Human Feedback (RLHF) for our generative question answering task, several key design considerations were crucial to ensuring the effectiveness and reliability of the process. These considerations encompassed dataset augmentation, model selection, human feedback integration, and iterative refinement. By carefully addressing these factors, we aimed to optimize the training and performance of our models, leveraging the strengths of RLHF to enhance the quality of generated answers.

4.3.1 Selection of Proper Dataset and Extraction of Required Data

For our work, we have selected the Cambridge Learner Corpus First Certificate in English (CLC FCE) dataset due to its alignment with our needs for question-answer pairs. This dataset contains short text answers written by English learners in response to exam prompts, providing a rich source of diverse questions and answers.

The dataset has number of questions, corresponding instructions, answers and obtained marks stored in several xml files. In early stage of our work, we have extracted these in an excel file with columns “Instruction”, “Question”,“Answer”,“Score” as required. We have also added a new column “Grade” in which we have assigned “Good” if corresponding “Score” is higher than or equal 28 and “Bad” otherwise. In each case total mark for each question was 40.

Table 4.1: Extracted Data Sample

Instruction	Question	Answer	Score	Grade
Read the following input	You have	Cambridge 13.06.0	24	Bad
Read the following input	You have	1ST June 2000 Dear	37	Good
Read the following input	You have	Dear Competition	28	Good
Read the following input	You have	Dear Helen Ryan ,	19	Bad
Read the following input	You have	Dear Helen Ryan:	29	Good

4.3.2 Selection of Proper LLM Models for Fine-tuning

In our thesis, we have employed two prominent Large Language Models (LLMs) for fine-tuning: GPT-2 and LLaMA-2. Both models were chosen for their robust capabilities in natural language processing and their suitability for generating coherent and contextually relevant text.

GPT-2: Developed by OpenAI, GPT-2 is renowned for its ability to generate high-quality text based on a given prompt. Its architecture allows it to handle a wide range of language tasks, making it an excellent choice for fine-tuning in the context of question answering. By leveraging its pre-trained capabilities, we were able to adapt GPT-2 to better respond to the specific types of questions and contexts present in the CLC FCE dataset.

LLaMA-2: The LLaMA-2 model, with its advanced transformer architecture, further complements our fine-tuning process. Known for its efficiency and accuracy, LLaMA-2 helps in enhancing the quality and relevance of generated answers. Its ability to process and generate text efficiently makes it a valuable asset in handling the diverse and complex inputs from the CLC FCE dataset.

By fine-tuning these models on our dataset, we have significantly improved their performance in generating accurate and contextually appropriate answers, leveraging the strengths of both GPT-2 and LLaMA-2 to meet the demands of our generative question answering task.

4.3.3 Performance Evaluation of Fine-tuned Model

We have evaluated the fine-tuned GPT-2 and LLaMA-2 models on 38 unique questions as obtained from the FCE dataset. The performance was assessed based on metrics like accuracy, relevance, fluency, and human-likeness. We have examined the improvement in models' ability to handle diverse and complex questions.

4.3.4 RLHF Reward Modeling

For our RLHF reward modeling, we used the RoBERTa-base model due to its strong performance in natural language understanding. The dataset was enhanced with two binary columns, ”Good” and ”Bad,” indicating the quality of answers with ”True” or ”False” based on human evaluation. The RoBERTa-base model was fine-tuned to predict these labels, optimizing it to generate a scalar reward representing answer quality. This reward was then used to guide the reinforcement learning process. Data preprocessing involved tokenizing the text and preparing the input format for RoBERTa. Continuous evaluation and iteration were performed to refine the model based on new feedback, enhancing its alignment with human preferences.

4.3.5 RLHF: Fine-Tuning with Reinforcement Learning

For the reinforcement learning aspect, we have adopted Proximal Policy Optimization (PPO), a robust policy-gradient algorithm well-suited for large-scale language models. The process began with the pre-trained GPT-2 and LLaMA-2 models, which were further fine-tuned using the scalar rewards derived from the RoBERTa model. This involved iterative training cycles where the models generated answers, received feedback in the form of ”Good” or ”Bad” labels, and adjusted their parameters to maximize the reward signal.

The reward function combined the predictions from the RoBERTa model with a constraint on policy shift, ensuring the models did not deviate too far from their pre-trained states, thus maintaining coherence and preventing gibberish output. The continuous cycle of generation, feedback, and adjustment allowed the models to learn and align more closely with human preferences, significantly enhancing the quality and relevance of their generated answers.

By integrating human feedback directly into the reinforcement learning loop, our implementation of RLHF successfully fine-tuned the generative models to produce higher-quality, contextually appropriate answers, demonstrating the effectiveness of this approach in practical applications.

4.3.6 Comparison of Performance before and after RLHF Integration

We evaluated the performance of our models on 38 unique questions from the FCE dataset before and after the integration of Reinforcement Learning from Human Feedback (RLHF).Using the reward model, we calculated reward values for the generated answers both before and after RLHF implementation. This comparative analysis aimed to determine the impact of RLHF on the models' performance by quantifying the improvement in reward values, thus reflecting the quality and contextual appropriateness of the generated responses.

4.4 Simulation and Experimental Methods

This section details the experimental method, data preparation, model training, simulation details, evaluation metrics, validation and testing methods used in our research.

4.4.1 Experimental Method

The experiments were conducted using Google Colab and Kaggle, providing the necessary computational resources for training and evaluating the models. These platforms facilitated efficient data handling and model training through their robust GPU support and integrated development environments.

4.4.2 Data Preparation

As previously discussed, the Cambridge Learner Corpus First Certificate in English (CLC FCE) dataset was utilized. This dataset includes short texts written by learners of English in response to exam prompts. The dataset was preprocessed to tokenize the text and split into training and testing sets.

4.4.3 Model Training

Key aspects of the training included:

1. Learning Rate: Adjusted the learning rate for efficient training.

2. Batch Size: Defined according to the computational limits of the environment.

3. Epochs: Multiple epochs were run to ensure adequate training.

4. Training Duration: Monitored to balance between model performance and computational efficiency.

4.4.4 Simulation Details and Evaluation Metrics

The simulations were designed to evaluate the models' performance through several metrics:

1. Mean Reward vs. Epoch: Tracking the average reward received by the model per epoch to assess learning progress.

2. Standard Deviation vs. Epoch: Measuring the variability in rewards to evaluate consistency.

3. Reward Value Before and After RLHF: Comparing the reward values to quantify the impact of RLHF.

4. Visual Comparison Before and After RLHF: Visual representations of model outputs before and after applying RLHF to illustrate improvements.

4.4.5 Validating and Testing

The full dataset comprises of 2,459 entries. Validation and testing were conducted using 20 percent of the dataset. This split ensured a robust evaluation of model performance.

4.4.6 Performance Evaluation

Additionally, performance was specifically evaluated on 38 unique questions to provide detailed insights into the models' capabilities.

The models were evaluated using the following criteria:

1. Accuracy: The correctness of generated answers.

2. Relevance: The contextual appropriateness of the responses.

3. Fluency: The grammatical and linguistic quality.

4. Human-likeness: The resemblance of generated answers to human responses.

Chapter 5

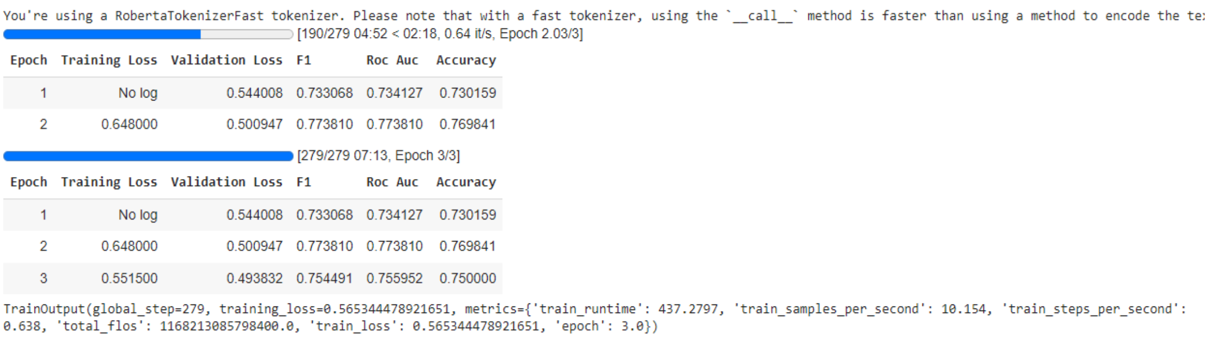
Results and Discussion

This chapter evaluates the performance of GPT-2 and LLaMA2-7B models on generative question answering with the integration of Reinforcement Learning from Human Feedback (RLHF). The models were assessed before and after Reinforcement Learning from Human Feedback (RLHF). It includes performance evaluation of reward model, RLHF on GPT-2, RLHF on LLaMA2-7B, and a performance comparison between the two models.This chapter also discusses on social-environmental impacts and ethical issues of the study.

5.1 Performance Analysis of Reward Model

We train RoBERTa Base to differentiate between good and bad answers and to provide a reward in return.

(i) The training was done in three epochs with a batch size of sixteen. Maximum accuracy was obtained on the second epoch.



```
You're using a RobertaTokenizerFast tokenizer. Please note that with a fast tokenizer, using the `__call__` method is faster than using a method to encode the te:
```

[190/279 04:52 < 02:18, 0.64 it/s, Epoch 2.03/3]

Epoch	Training Loss	Validation Loss	F1	Roc Auc	Accuracy
1	No log	0.544008	0.733068	0.734127	0.730159
2	0.648000	0.500947	0.773810	0.773810	0.769841

[279/279 07:13, Epoch 3/3]

Epoch	Training Loss	Validation Loss	F1	Roc Auc	Accuracy
1	No log	0.544008	0.733068	0.734127	0.730159
2	0.648000	0.500947	0.773810	0.773810	0.769841
3	0.551500	0.493832	0.754491	0.755952	0.750000

```
TrainOutput(global_step=279, training_loss=0.565344478921651, metrics={'train_runtime': 437.2797, 'train_samples_per_second': 10.154, 'train_steps_per_second': 0.638, 'total_flos': 1168213085798400.0, 'train_loss': 0.565344478921651, 'epoch': 3.0})
```

Figure 5.1: Reward Model Training Logs

Accuracy on validation data was about 78 percent as we obtained.

```
[16/16 00:07]
{'eval_loss': 0.5009470582008362,
 'eval_f1': 0.7738095238095238,
 'eval_roc_auc': 0.7738095238095237,
 'eval_accuracy': 0.7698412698412699,
 'eval_runtime': 7.6111,
 'eval_samples_per_second': 33.11,
 'eval_steps_per_second': 2.102,
 'epoch': 3.0}
```

Figure 5.2: Validation Metrics

(ii) Now, as we infer with a set of questions and corresponding answers, the trained model provides us the predicted labels. In each case predicted label, probability of being "Good" and probability of being "Bad" are shown along with original label indications. Observation suggests that the model is able to predict the expected label in maximum cases (here, 17 out of 18 cases).

```
0 ['Bad'] False 0.42126139998435974 True 0.5679137706756592
1 ['Bad'] False 0.13291577994823456 True 0.861589789390564
2 ['Good'] True 0.6130512356758118 False 0.39301052689552307
3 ['Good'] True 0.8955992460250854 False 0.10170872509479523
4 ['Good'] True 0.5287973880767822 False 0.47924748063087463
5 ['Bad'] False 0.13405565917491913 True 0.8595477342605591
6 ['Good'] True 0.7713487148284912 False 0.2324124574661255
7 ['Good'] True 0.6432353854179382 False 0.35768720507621765
8 ['Good'] True 0.9132167100906372 False 0.09170974045991898
9 ['Good'] True 0.8066664338111877 False 0.21069933474063873
10 ['Good'] True 0.7917518019676208 False 0.21807551383972168
11 ['Bad'] False 0.11884059756994247 True 0.8751915693283081
12 ['Bad'] False 0.18654179573059082 True 0.8040432929992676
13 ['Good'] True 0.8746283054351807 False 0.12817110121250153
14 ['Good'] True 0.8118165135383606 False 0.18157172203063965
15 ['Bad'] False 0.20386061072349548 True 0.791178822517395
16 ['Bad'] False 0.19205191731452942 True 0.7969245314598083
17 ['Good'] False 0.6889536380767822 True 0.3224804401397705
18 ['Good'] True 0.9402937889099121 False 0.06327469646930695
```

Figure 5.3: Prediction Using Reward Model

Calling the trained model through a pipeline returns predicted label, reward score and corresponding logits.

```
[14] qa_pipe('Write an article suggesting good ways for young people to earn some money in their spare time.',
        second_text='In the last few years, the number of young professionals has increased.  This is because of the   increasing number  of jobs available in tl

{'label': 'Good',
 'score': 0.989443838596344,
 'logits': [-2.2725422382354736, 2.2678911685943604]}
```

Figure 5.4: Evaluation of a Good Answer

```
qa_pipe('Write a composition on "food and eating habits.',
        second_text='Food and Eating Habits  Food is the most important thing in our life.But We can live without it. But what kind of food do we eat? What is oı

{'label': 'Bad',
 'score': 0.6506581902503967,
 'logits': [0.25692638754844666, -0.3650071322917938]}
```

Figure 5.5: Evaluation of a Bad Answer

We utilize these rewards while performing RLHF on LLMs.

5.2 RLHF on GPT-2

5.2.1 Tracking Learning Progress and Consistency:

The graph below illustrates the mean reward and standard deviation of rewards over multiple epochs during the training of the GPT-2 model with Reinforcement Learning from Human Feedback (RLHF).

Mean Reward Trend:

i. The mean rewards (blue line) show a clear upward trend as the epochs progress. Initially, the rewards start below zero, indicating suboptimal performance.

ii. By the third epoch, there is a significant increase in mean rewards, reaching approximately 1.5. This indicates that the model is learning effectively and adapting to the feedback.

iii. The mean reward continues to increase, peaking around the sixth epoch at approximately 2.25, demonstrating continued improvement and optimization of the model.

iv. The slight fluctuations observed after the peak suggest minor adjustments as the model fine-tunes its responses, but overall, the trend remains positive, stabilizing above 2.0.

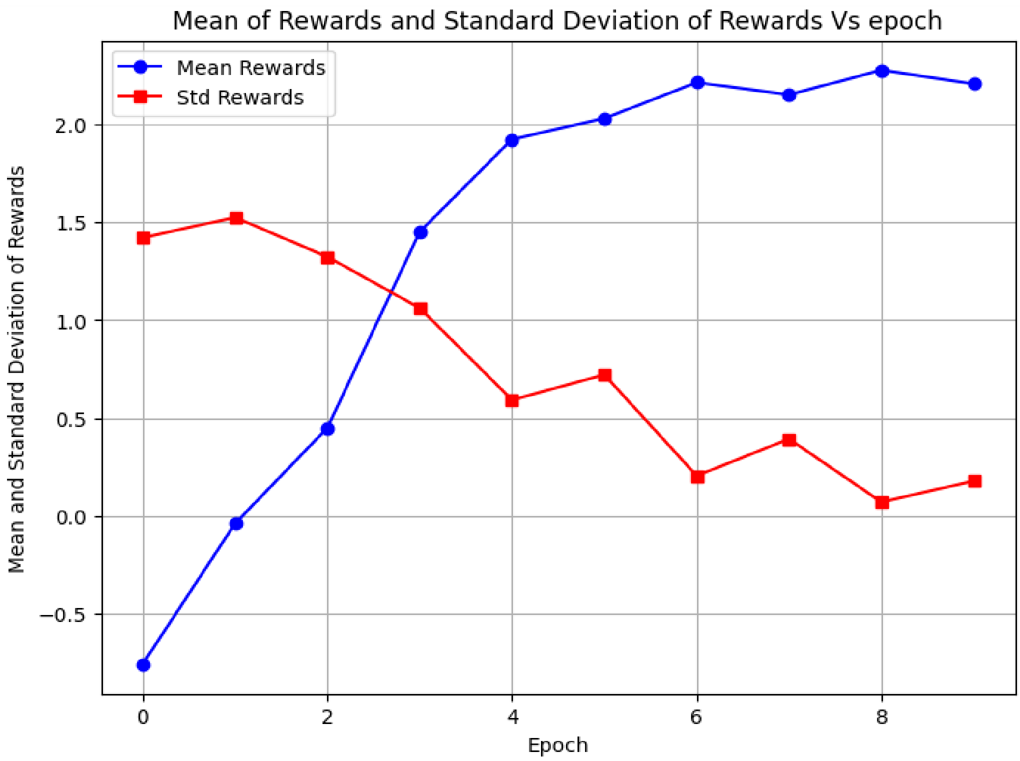


Figure 5.6: Tracking Learning Progress and Consistency (GPT-2)

Standard Deviation Trend:

i. The standard deviation of rewards (red line) starts relatively high, around 1.5, indicating variability in the model's performance across different samples.

ii. As training progresses, the standard deviation begins to decrease, with a noticeable drop around the fourth epoch. This decrease indicates that the model's performance is becoming more consistent and reliable.

iii. By the sixth epoch, the standard deviation falls below 0.25, showing a significant reduction in variability.

iv. Although there are minor fluctuations in the standard deviation after this point, the overall trend remains low, maintaining consistency in the model's performance.

Overall, the graphs demonstrate that the integration of RLHF significantly improves the GPT-2 model's learning progress, as evidenced by the rising mean rewards. Additionally, the decreasing standard deviation indicates enhanced consistency and reliability in the model's performance over time.

5.2.2 Reward Value Distribution Before and After RLHF Implementation:

Evaluating Reward Distribution for Training Data:

1. Fine Tuned GPT-2:

```
Statistical measures for Gen_Reward(Training Data):
Mean: 0.3420366163437183
Variance: 2.7488562726157193
Median: 0.4633893221616745
```

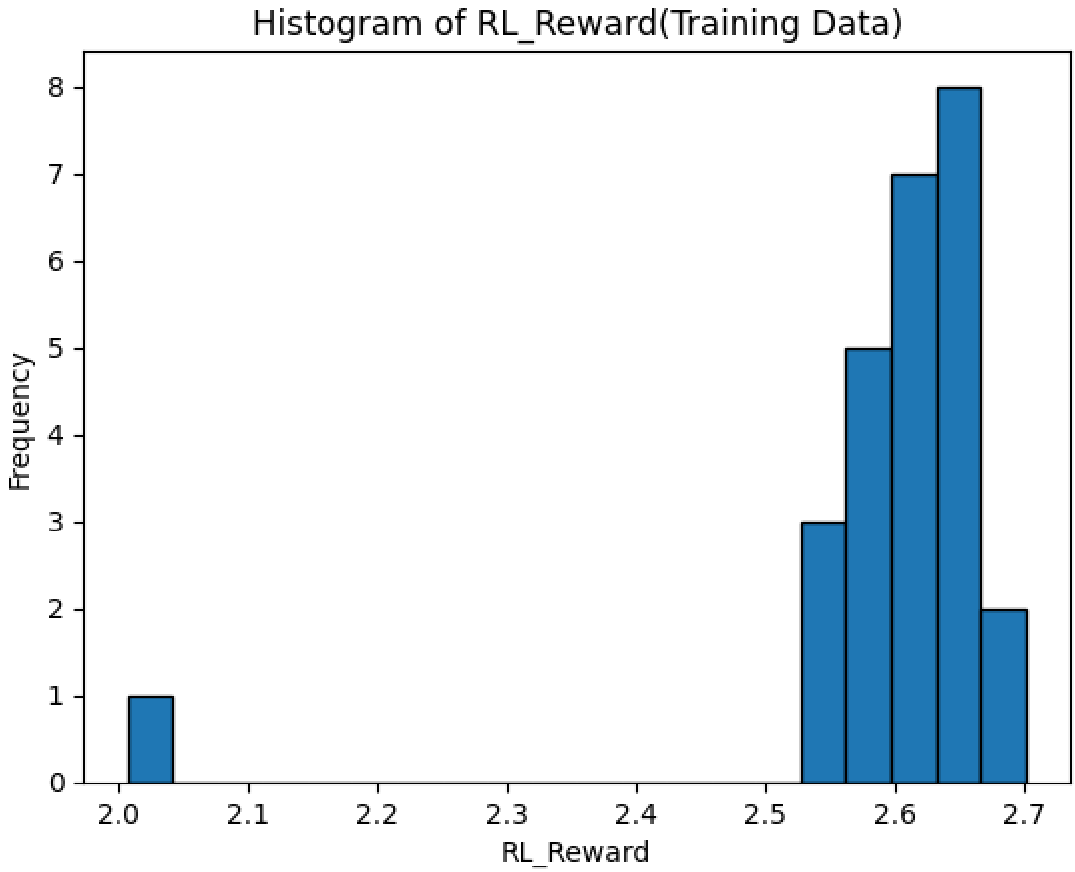

```
Number of 'Good' in Gen_Grade column: 15
Percentage of 'Good' in Gen_Grade column: 57.692307692307686
```

Figure 5.7: Fine-Tuned GPT-2 Reward Statistics

The histogram shows a broad distribution of reward values ranging from -2 to 3, indicating variability in the fine tuned model's performance. Approximately 57.69 percent of generated sample answers were labeled as "Good" by the reward model.

2. RL Reward Distribution:

```
Statistical measures for RL_Reward(Training Data):
Mean: 2.5946730283590465
Variance: 0.016181014210479728
Median: 2.6141791343688965
```




```
Number of 'Good' in RL_Answer column: 26
Percentage of 'Good' in RL_Answer column: 100.0
```

Figure 5.8: GPT-2 Reward Statistics after Implementing RLHF

The histogram shows a concentrated distribution around 2.6, indicating a significant improvement in the model's performance after RLHF. 100 percent of generated sample answers were labeled as “Good” by the reward model.

Evaluating Reward Distribution for Validation Data:

1. Fine Tuned GPT-2:

```
Statistical measures for Gen_Reward(Evaluation Data):
Mean: 0.550717880949378
Variance: 2.196339870755388
Median: 0.49634793773293495
```

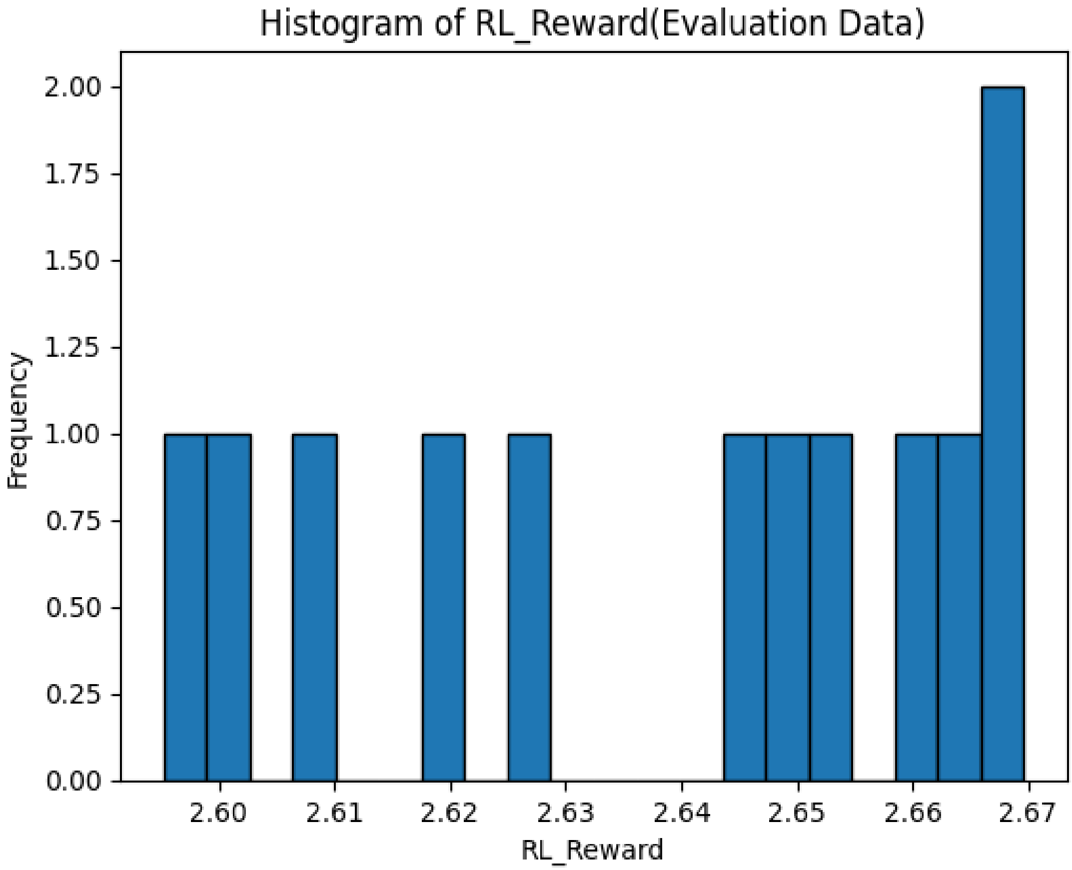

```
Number of 'Good' in Gen_Grade column: 6
Percentage of 'Good' in Gen_Grade column: 50.0
```

Figure 5.9: Fine-Tuned GPT-2 Reward Statistics

The histogram shows a broad distribution of reward values ranging from -2 to 3. Only 50 percent of generated sample answers were labeled as “Good” by the reward model.

2. RL Reward Distribution:

```
Statistical measures for RL_Reward(Evaluation Data):
Mean: 2.638237774372101
Variance: 0.0007154565884622922
Median: 2.647055745124817
```




```
Number of 'Good' in RL_Answer column: 12
Percentage of 'Good' in RL_Answer column: 100.0
```

Figure 5.10: GPT-2 Reward Statistics after Implementing RLHF

The histogram demonstrates a very narrow distribution from 2.6 to 2.67, reflecting consistent high performance on the evaluation data after RLHF. 100 percent of generated sample answers were labeled as "Good" by the reward model.

The reward distributions before and after RLHF show a dramatic improvement in the model's performance. Before RLHF, the rewards were widely distributed with significant variability, and only about half of the generated answers were considered "Good." After applying RLHF, the rewards concentrated around higher values with reduced variance, and all samples were labeled as "Good." This indicates that RLHF effectively enhanced the model's ability to generate high-quality, reliable responses consistently.

The results are shown in the table below for comparison.

Table 5.1: GPT-2 Reward Value Distribution

	Training Data		Validation Data	
	After FT	After RLHF	After FT	After RLHF
Mean Reward	0.342	2.595	0.5507	2.638
Variance	2.749	0.016	2.196	0.0007
Range	-2 to 3	2 to 2.7	-2 to 3	2.6 to 2.67
% Good Response	57.69%	100%	50%	100%

5.2.3 Visual Comparison Before and After RLHF:

Example 1:

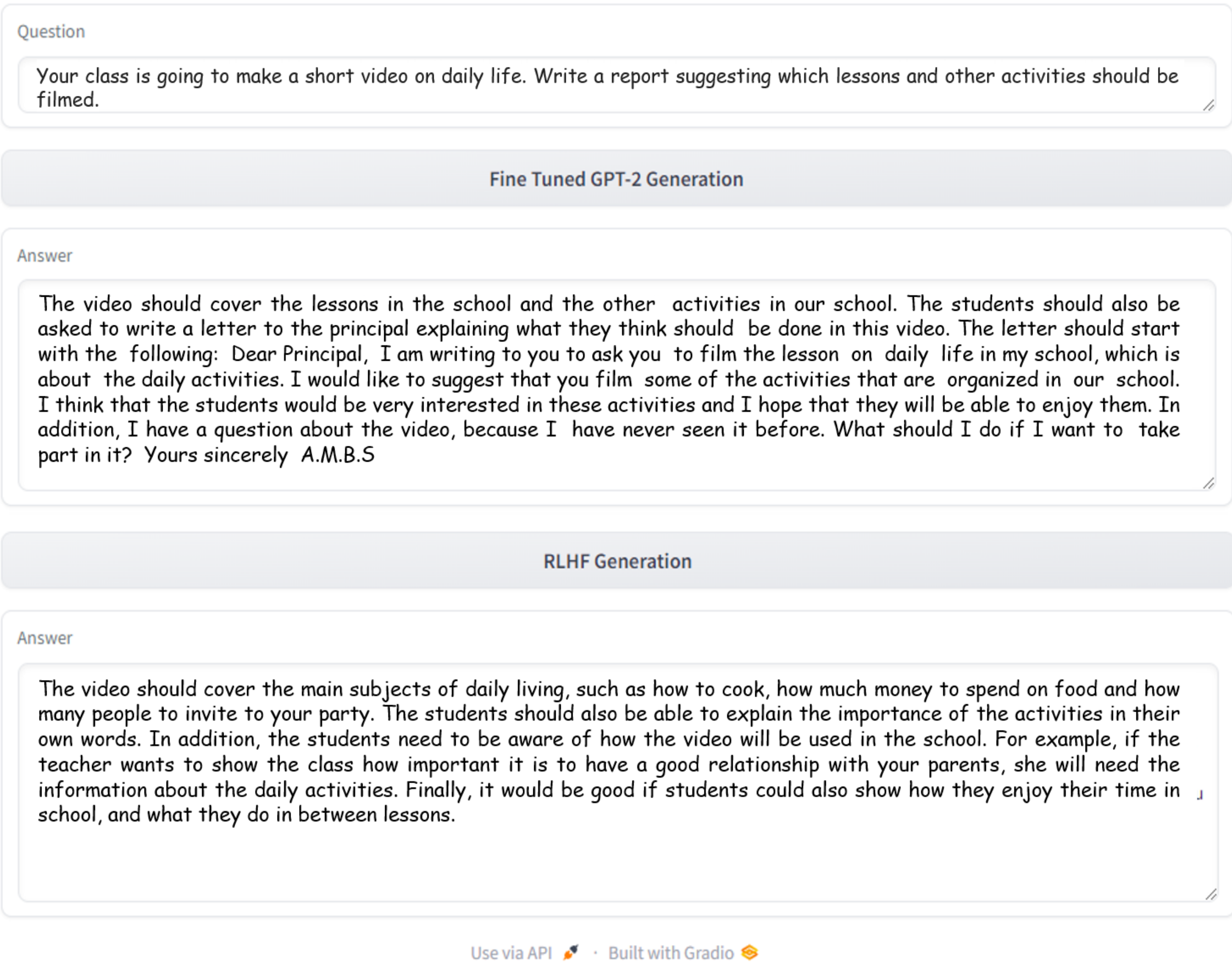


Figure 5.11: Visual Comparison for GPT-2 (1)

Example 2:

Figure 5.12: Visual Comparison for GPT-2 (2)

Example 3:

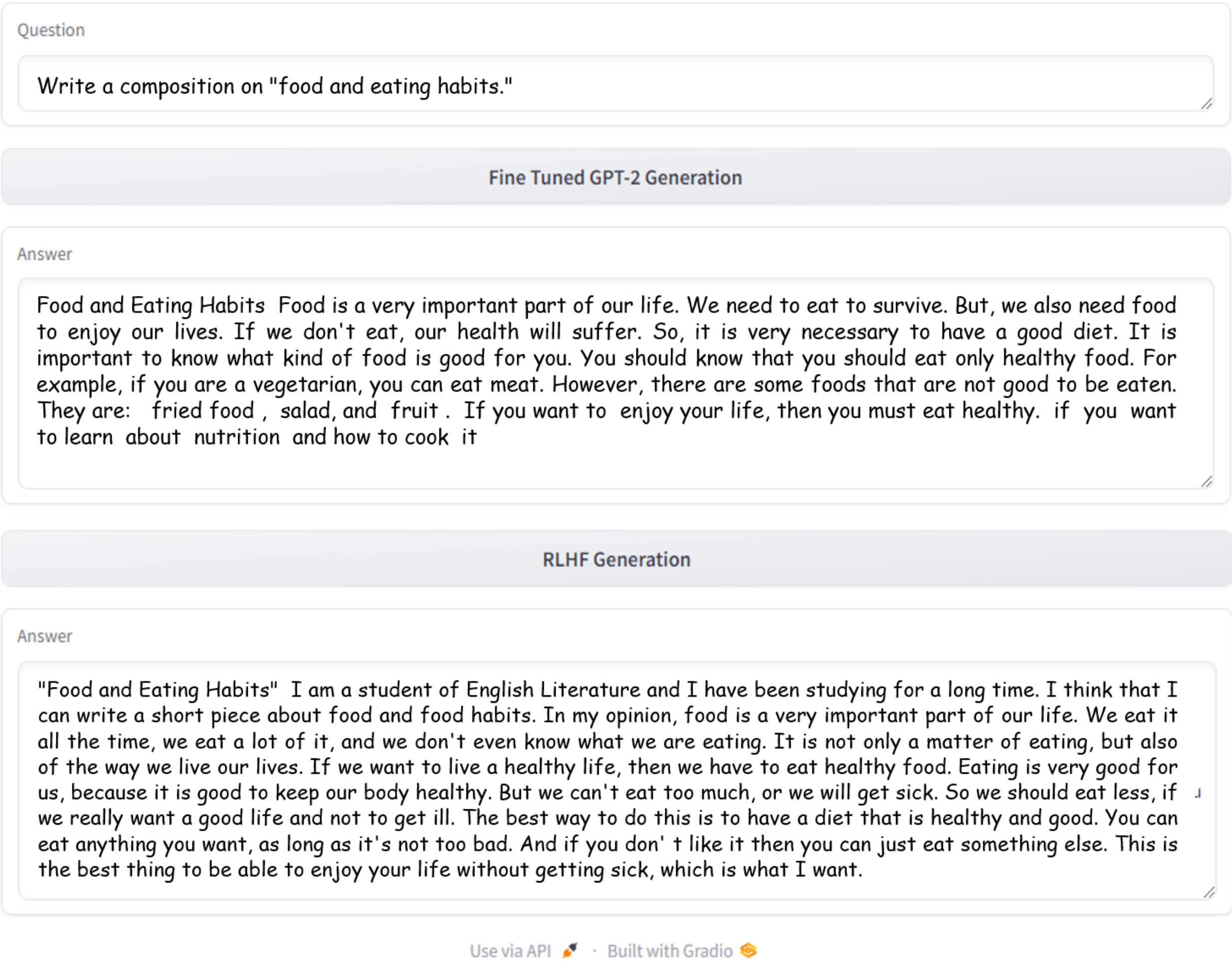


Figure 5.13: Visual Comparison for GPT-2 (3)

Above examples provide a comparative analysis between the Fine-Tuned GPT-2 Generation and Generation after RLHF Implementation. Observations from the images highlight the following points:

Clarity and Relevance: Both Fine-Tuned GPT-2 and RLHF Implemented GPT-2 models generate responses that are relevant to the given questions. However, RLHF responses tend to be more coherent and directly address the query with better structure and clarity.

Answer Quality: The Fine-Tuned GPT-2 model produces answers that are somewhat repetitive and lack a natural flow. The RLHF model, on the other hand, provides more detailed and human-like responses that are less redundant and more engaging.

Contextual Understanding: RLHF generations show a deeper understanding of the context. For instance, in the first image, the RLHF model suggests specific activities and emphasizes the importance of understanding daily life, which is more nuanced compared to the Fine-Tuned GPT-2's output.

Fluency and Grammar: The grammar and fluency of the RLHF model's responses are superior. The sentences are well-formed and follow a logical progression, making the text easier to read and understand.

User Engagement: Responses from the RLHF model are more likely to engage the user due to their detailed, informative, and contextually appropriate nature. For example, the RLHF response in the second image includes specific attractions and suggests a practical solution (arranging a free ticket), which adds value to the answer.

In summary, while the Fine-Tuned GPT-2 model is capable of generating relevant answers, the RLHF model significantly enhances the quality, coherence, and human-likeness of the responses, making it more effective for generative question answering tasks.

5.3 RLHF on LLaMA2-7B

5.3.1 Tracking Learning Progress and Consistency:

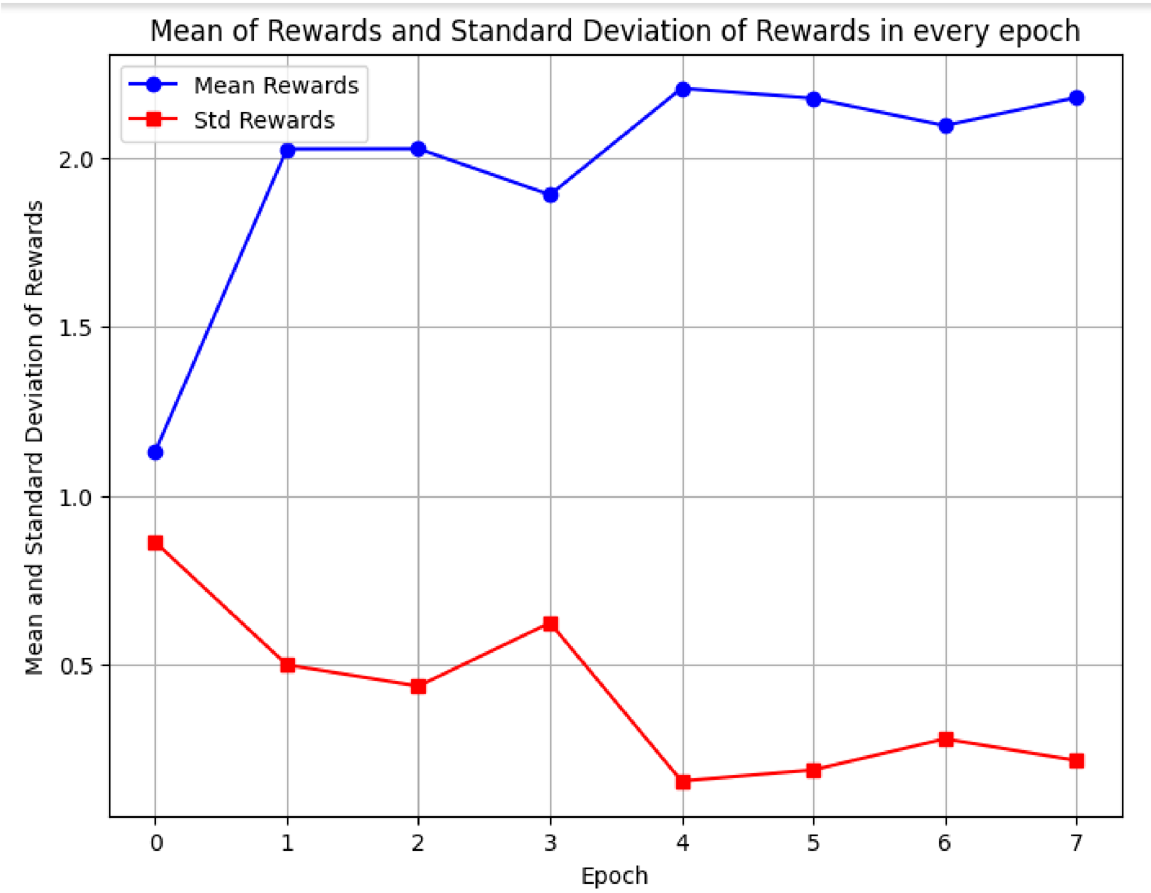


Figure 5.14: Tracking Learning Progress and Consistency (LLaMA2-7B)

The graph above illustrates the mean reward and standard deviation of rewards over multiple epochs during the training of the LLaMA2-7B model with Reinforcement Learning from Human Feedback (RLHF).

Mean Reward Trend:

i. The mean rewards (blue line) show a clear upward trend as the epochs progress. Initially, the rewards start below 1.25, indicating suboptimal performance.

ii. By the second epoch, there is a significant increase in mean rewards, reaching approximately 2.0. This indicates that the model is learning effectively and adapting to the feedback.

iii. The mean reward continues to increase, peaking around the fourth epoch at approximately 2.2, demonstrating continued improvement and optimization of the model.

iv. The slight fluctuations observed after the peak suggest minor adjustments as the model fine-tunes its responses, but overall, the trend remains positive, stabilizing above 2.0.

Standard Deviation Trend:

i. The standard deviation of rewards (red line) starts relatively high, around 1.0, indicating variability in the model's performance across different samples.

ii. As training progresses, the standard deviation begins to decrease, with a noticeable drop around the fourth epoch. This decrease indicates that the model's performance is becoming more consistent and reliable.

iii. In the fourth epoch, the standard deviation falls below 0.1, showing a significant reduction in variability.

iv. Although there are minor fluctuations in the standard deviation after this point, the overall trend remains low, maintaining consistency in the model's performance.

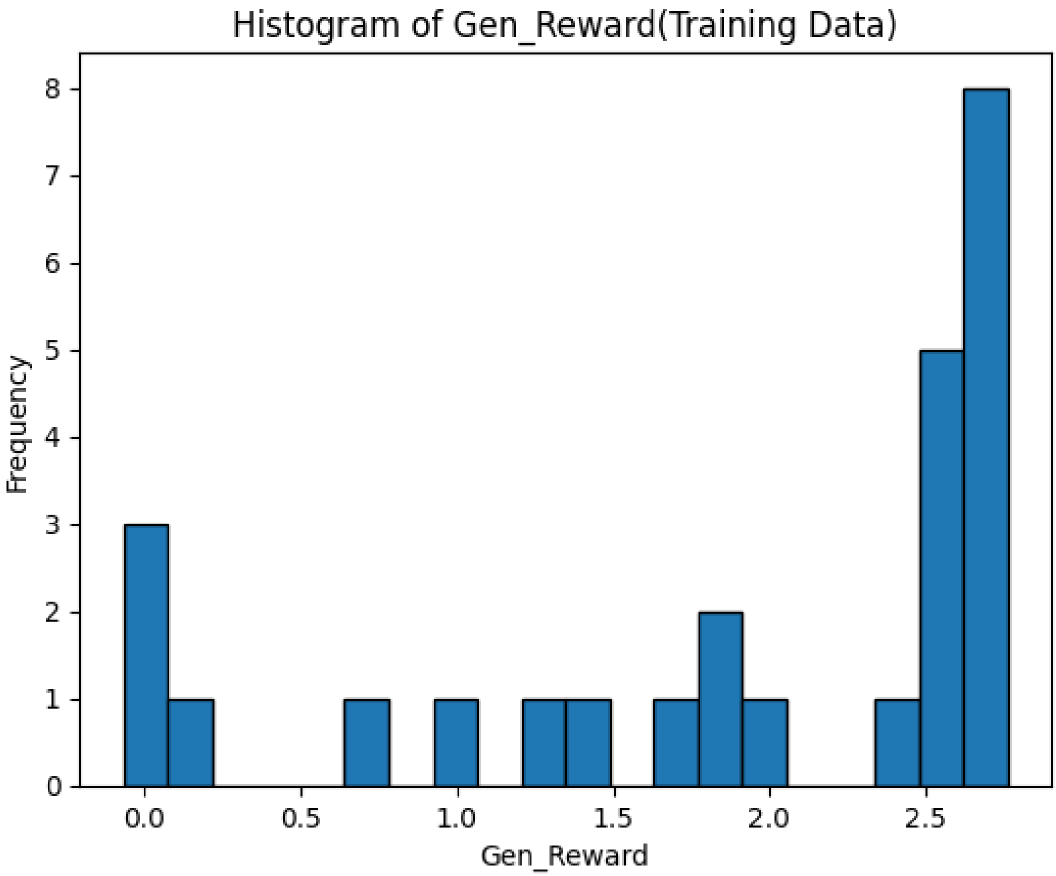
Overall, the graphs demonstrate that the integration of RLHF significantly improves the LLaMA2-7B model's learning progress, as evidenced by the rising mean rewards. Additionally, the decreasing standard deviation indicates enhanced consistency and reliability in the model's performance over time.

5.3.2 Reward Value Distribution Before and After RLHF Implementation:

Evaluating Reward Distribution for Training Data:

1. Fine Tuned LLaMA2-7B:

```
Statistical measures for Gen_Reward(Training Data):
Mean: 1.8780085829874644
Variance: 0.9784326861771306
Median: 2.4937973022460938
```



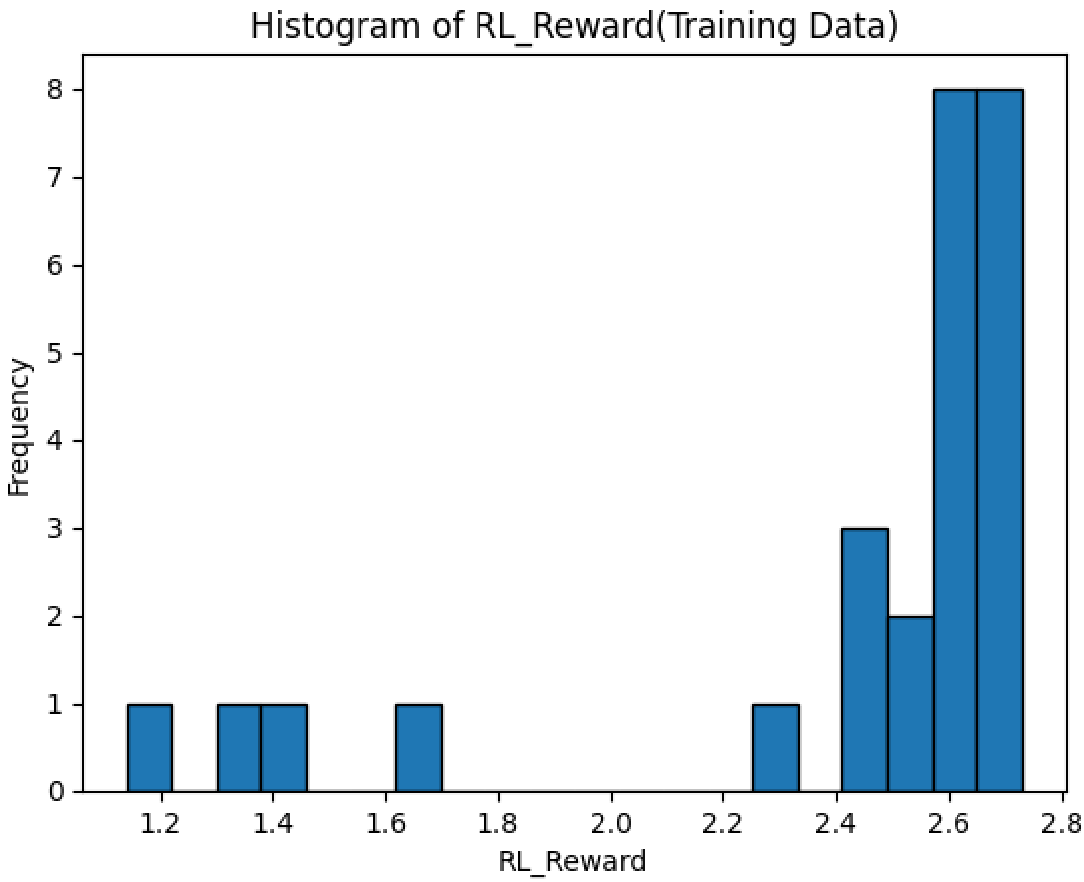

```
Number of 'Good' in Gen_Grade column: 24
Percentage of 'Good' in Gen_Grade column: 92.3076923076923
```

Figure 5.15: Fine-Tuned LLaMA2-7B Reward Statistics

The histogram shows a broad distribution of reward values ranging from 0 to 2.7, indicating variability in the fine-tuned model's performance. Approximately 92.31 percent of generated sample answers were labeled as "Good" by the reward model.

2. RL Reward Distribution:

```
Statistical measures for RL_Reward(Training Data):
Mean: 2.4110154463694644
Variance: 0.2165255627898071
Median: 2.610270619392395
```




```
Number of 'Good' in RL_Answer column: 26
Percentage of 'Good' in RL_Answer column: 100.0
```

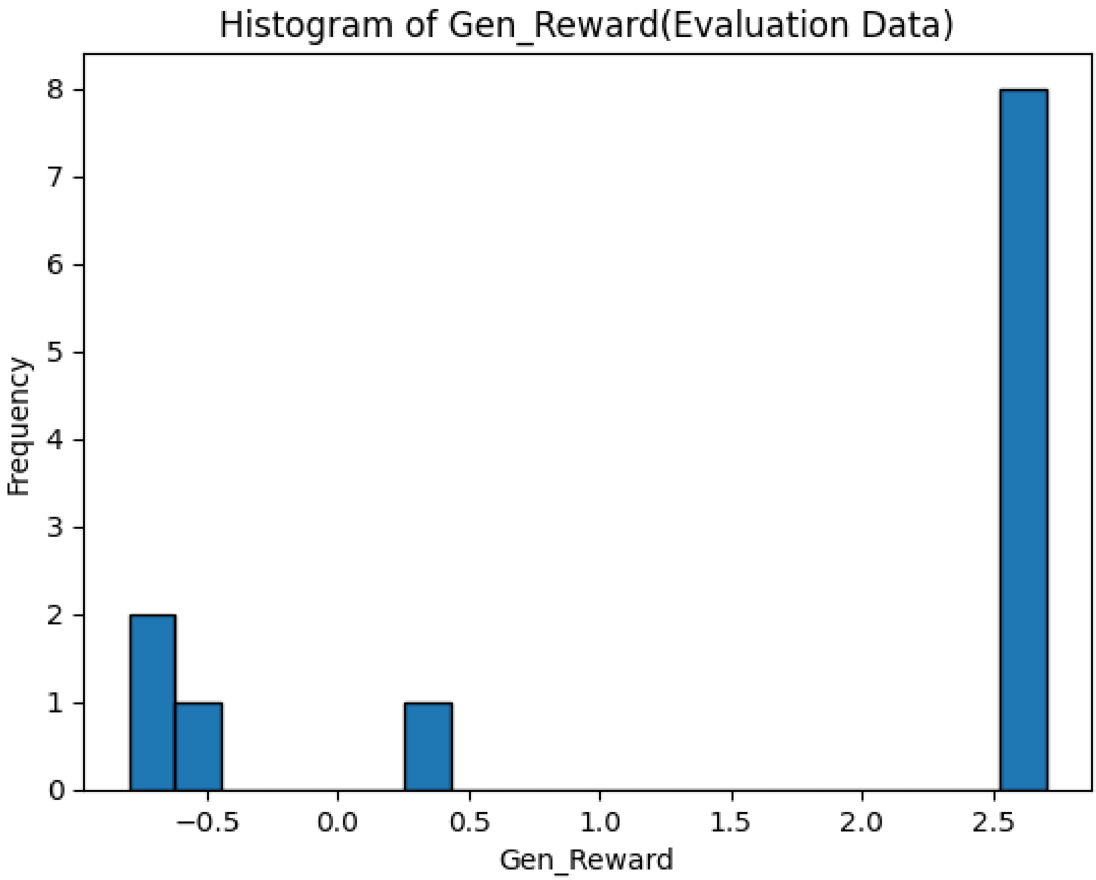
Figure 5.16: LLaMA2-7B Reward Statistics after Implementing RLHF

The histogram shows a concentrated distribution around 2.6, indicating a significant improvement in the model's performance after RLHF. 100 percent of generated sample answers were labeled as "Good" by the reward model.

Evaluating Reward Distribution for Validation Data

1. Fine Tuned LLaMA2-7B:

```
Statistical measures for Gen_Reward(Evaluation Data):
Mean: 1.6146558746695518
Variance: 2.3340162533001765
Median: 2.5857075452804565
```



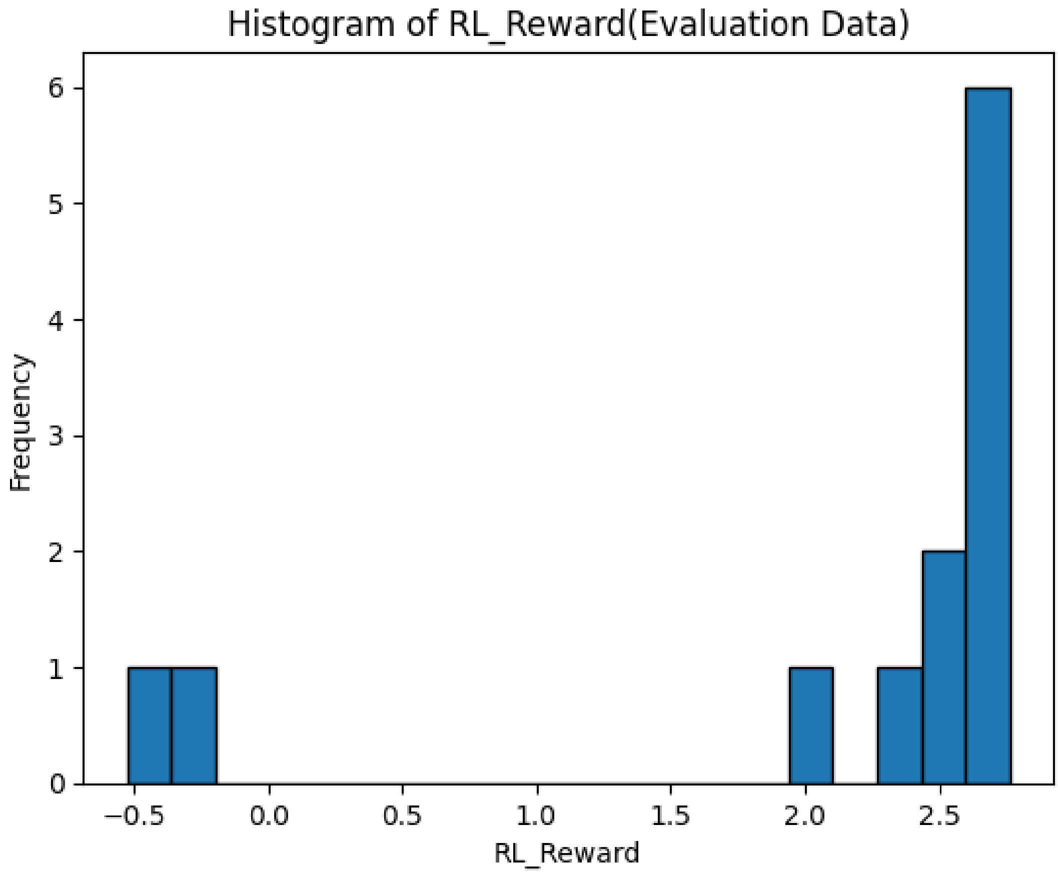

```
Number of 'Good' in Gen_Grade column: 9
Percentage of 'Good' in Gen_Grade column: 75.0
```

Figure 5.17: Fine-Tuned LLaMA2-7B Reward Statistics

The histogram shows a broad distribution of reward values ranging from -0.75 to 2.6. Only 75 percent of generated sample answers were labeled as “Good” by the reward model.

2. RL Reward Distribution:

```
Statistical measures for RL_Reward(Evaluation Data):
Mean: 2.0353892420729003
Variance: 1.3412068318211492
Median: 2.539010763168335
```




```
Number of 'Good' in RL_Answer column: 10
Percentage of 'Good' in RL_Answer column: 83.33333333333334
```

Figure 5.18: LLaMA2-7B Reward Statistics after Implementing RLHF

The histogram demonstrates a very narrow distribution around 2.5 to 2.65, reflecting consistent high performance on the evaluation data after RLHF. 83.33 percent of generated sample answers were labeled as "Good" by the reward model.

The reward distributions before and after RLHF show a dramatic improvement in the model's performance. After applying RLHF, the rewards concentrated around higher values with reduced variance, and a higher percentage of samples were labeled as "Good." This indicates that RLHF effectively enhanced the model's ability to generate high-quality, reliable responses consistently.

The results are shown in the table below for comparison.

Table 5.2: LLaMA2-7B Reward Value Distribution

	Training Data		Validation Data	
	After FT	After RLHF	After FT	After RLHF
Mean Reward	1.878	2.411	1.6147	2.0354
Variance	0.978	0.216	2.334	1.341
Range	0 to 2.7	1.2 to 2.7	-0.75 to 2.6	-0.5 to 2.65
% Good Response	92.31%	100%	75%	83.33%

5.3.3 Visual Comparison Before and After RLHF:

Example 1:

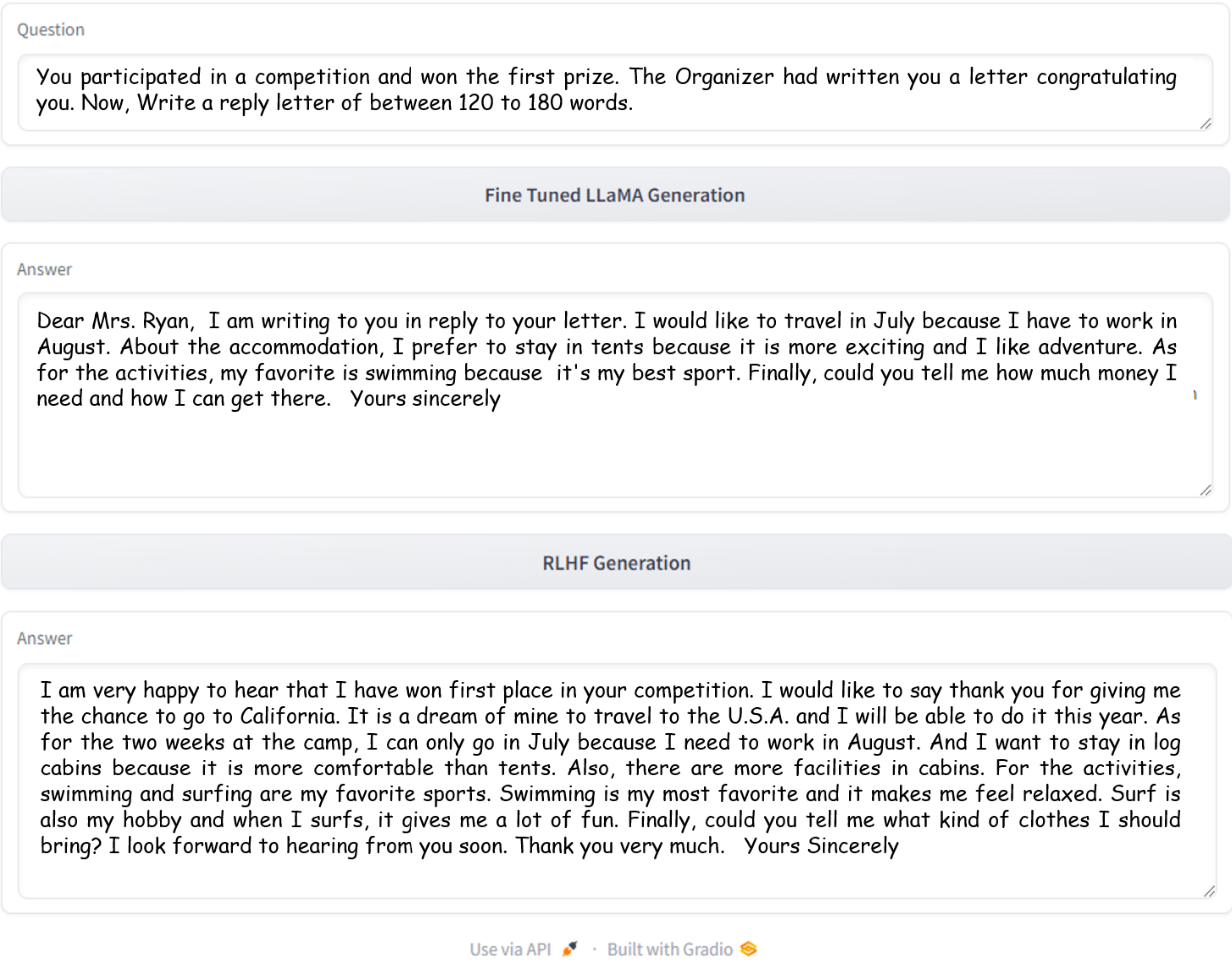


Figure 5.19: Visual Comparison for LLaMA2-7B (1)

Example 2:

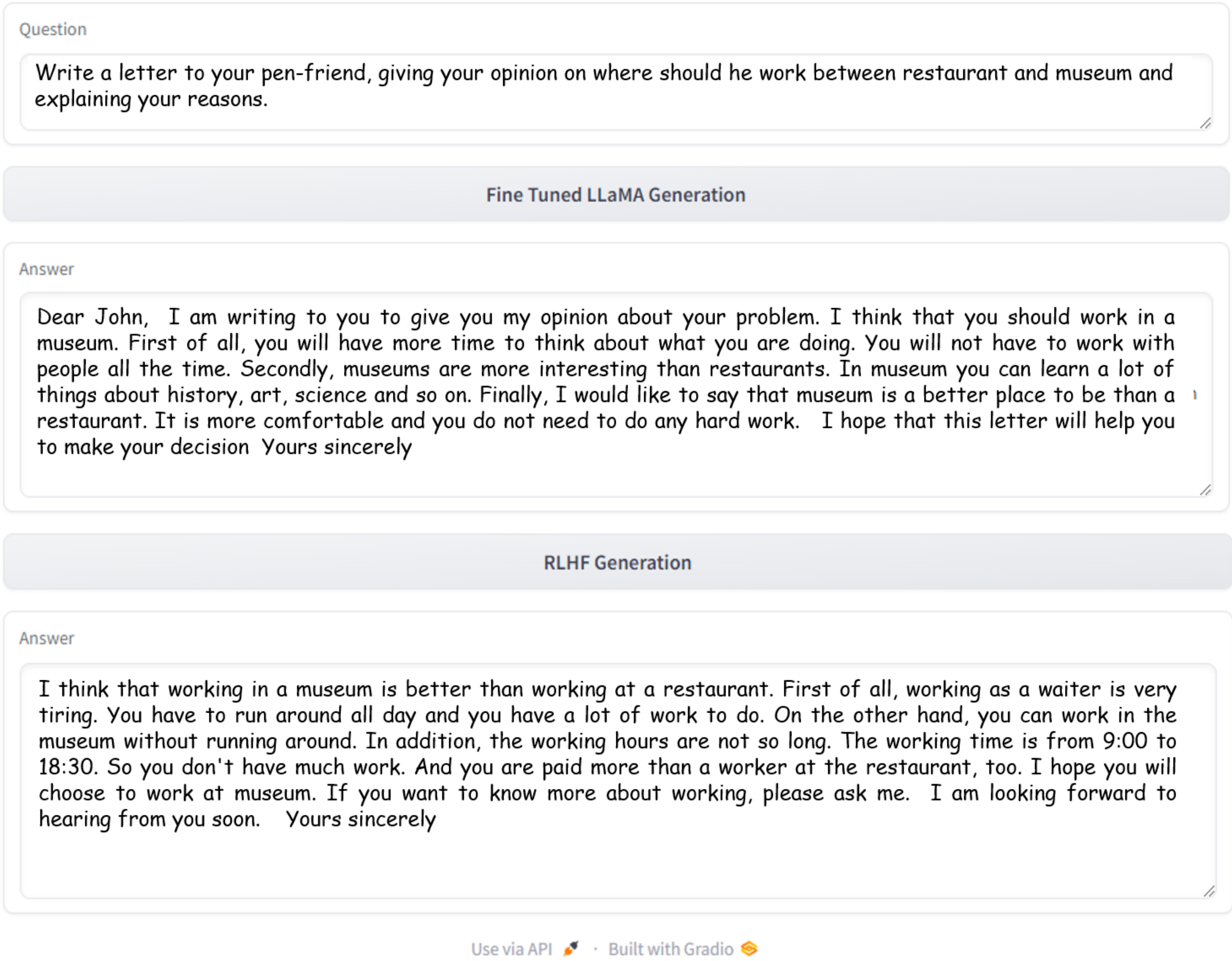


Figure 5.20: Visual Comparison for LLaMA2-7B (2)

Example 3:

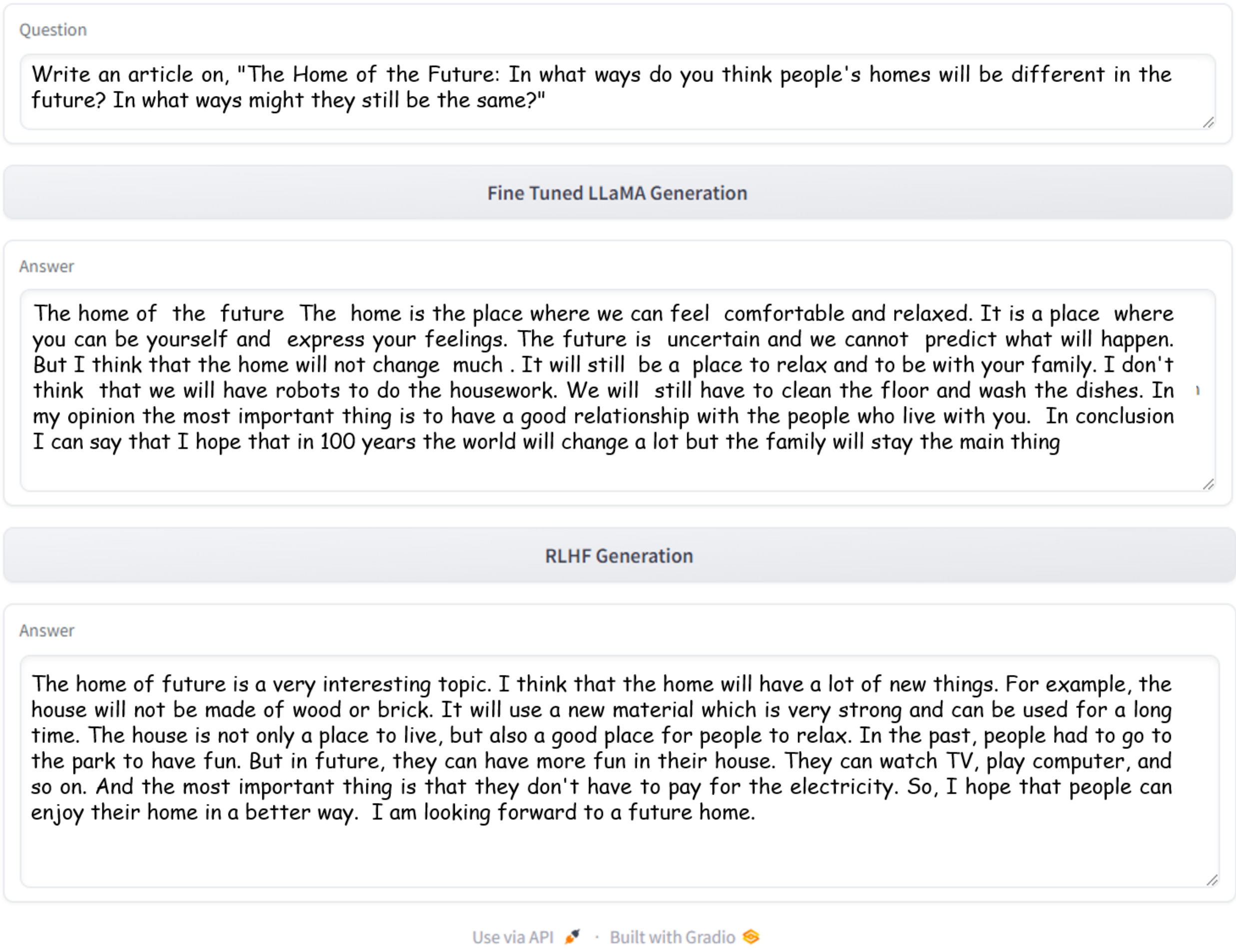


Figure 5.21: Visual Comparison for LLaMA2-7B (3)

Above examples provide a comparative analysis between the Fine-Tuned LLaMA2-7B Generation and Generation after RLHF Implementation. Observations from the images highlight the following points:

Clarity and Relevance: The responses from the Fine-Tuned LLaMA2-7B model are relevant to the given questions but tend to be straightforward and less detailed. The answers are generally on point but lack depth.The RLHF model generates responses that are not only relevant but also more comprehensive. The answers provide additional context and details that enhance the overall relevance and make them more useful.

Answer Quality: The answers from the Fine-Tuned LLaMA2-7B model are clear but often lack the richness and elaboration found in human-like responses. They provide basic information without delving into nuanced details.The RLHF model's responses are more elaborate and exhibit a natural flow. They include more specifics and contextual information, making the answers richer and more engaging. For instance, the RLHF responses suggest practical actions, provide additional information, and are more likely to fulfill the user's expectations.

Contextual Understanding: The Fine-Tuned responses show a basic understanding of the context but do not always capture the subtleties. The answers are often more generic. The RLHF model demonstrates a deeper understanding of the context. It includes detailed explanations and relevant examples that show a better grasp of the nuances. For example, in the first image, the RLHF response provides specific reasons and suggestions, which reflect a more profound contextual awareness.

Fluency and Grammar: The grammar and fluency of the Fine-Tuned model are adequate but can sometimes be simplistic and less sophisticated. The RLHF model produces responses with superior grammar and fluency. The sentences are well-structured and exhibit a logical progression, making the text more readable and professional.

User Engagement: The responses from the Fine-Tuned LLaMA2-7B model are functional but less engaging. They provide the necessary information but do not go beyond to captivate the reader. The RLHF model's responses are more likely to engage the user due to their detailed, informative, and contextually rich nature. They offer specific advice, additional context, and a more personable tone, making them more appealing and engaging.

Specific Comparisons:

Example 1:

Fine-Tuned LLaMA2-7B: Provides a basic, polite reply but lacks specific details and personal touches.
RLHF: Offers a detailed, enthusiastic reply, mentioning specific activities and showing gratitude, making it more engaging and human-like.

Example 2:

Fine-Tuned LLaMA2-7B: Provides clear reasons but is somewhat generic.
RLHF: Gives a more detailed and practical comparison, mentioning specific benefits and working conditions, making it more informative and useful.

Example 3:

Fine-Tuned LLaMA2-7B: Offers a basic prediction with limited details.
RLHF: Provides a vivid and imaginative description of future homes, including specific technological advancements and lifestyle changes, making it more compelling.

While the Fine-Tuned LLaMA2-7B model is capable of generating relevant and coherent responses, the RLHF model significantly enhances the quality, detail, and human-likeness of the answers. The RLHF responses are more engaging, contextually rich, and exhibit a better understanding of the nuances, making them more effective for generative question answering tasks.

5.4 Performance Comparison between GPT-2 and LLaMA2-7B upon RLHF Integration

5.4.1 Statistical Comparison

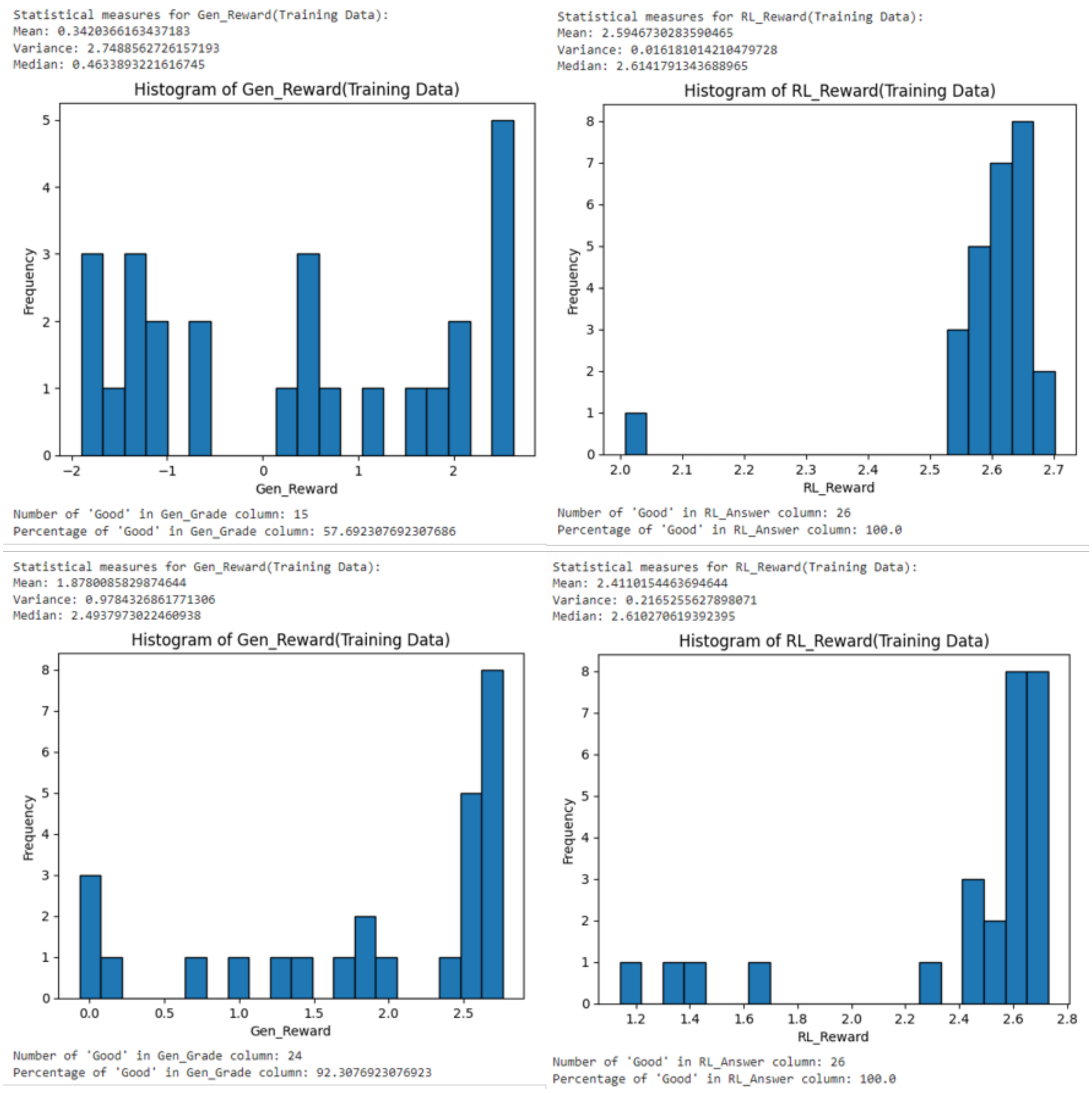


Figure 5.22: Performance Comparison between GPT-2 and LLaMA2-7B

If we compare the reward values after fine-tuning for both models, it is visible that LLaMa2-7B has positive reward values over a range from 0 to 2.6 with most occurrence around higher rewards. GPT-2 on the other hand has a wider distribution including negative reward values as well. This proves that LLaMA2-7B usually provides better and more human like answer than GPT-2 upon fine-tuning. Next, RLHF improves the rewards for both the models, resulting in a distribution centered around 2.6-2.7. If we talk about relative improvement, it is evident that GPT-2 achieves it more as it's responses were not that good initially. Additionally, LLaMa2-

7B also ensures high counts around peak reward value, resulting in 100 percent of generated answers are evaluated as “Good”.

5.4.2 Visual Comparison

Fine-Tuned Models:

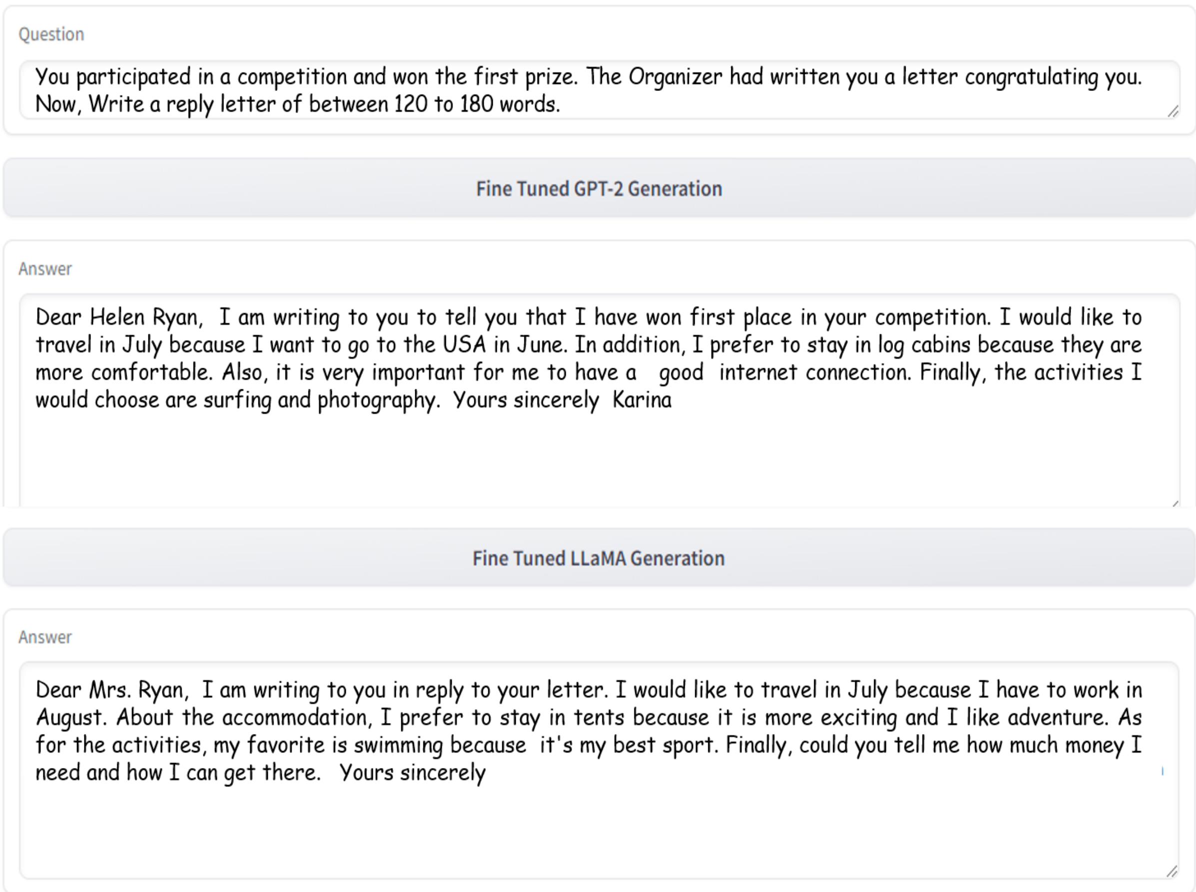


Figure 5.23: Visual Comparison between Fine Tuned Models

The comparison between the Fine-Tuned GPT-2 and Fine-Tuned LLaMA2-7B responses for the provided question reveals a clear advantage for the LLaMA2-7B model in terms of clarity, relevance, and human-like engagement. The LLaMA2-7B response is concise and straightforward, addressing the query with a natural flow that closely mimics human conversation. It highlights specific preferences such as the excitement of staying in tents and the enjoyment of favorite activities like swimming, which adds a personal touch to the response. Moreover, the LLaMA2-7B model maintains coherence and relevance throughout the answer, providing a well-rounded and engaging reply. In contrast, the GPT-2 response, while detailed, tends to be slightly verbose and repetitive. Overall, the LLaMA2-7B model excels in delivering a response that is not only relevant and coherent but also more relatable and engaging, making it a superior choice.

RLHF Implemented Models:

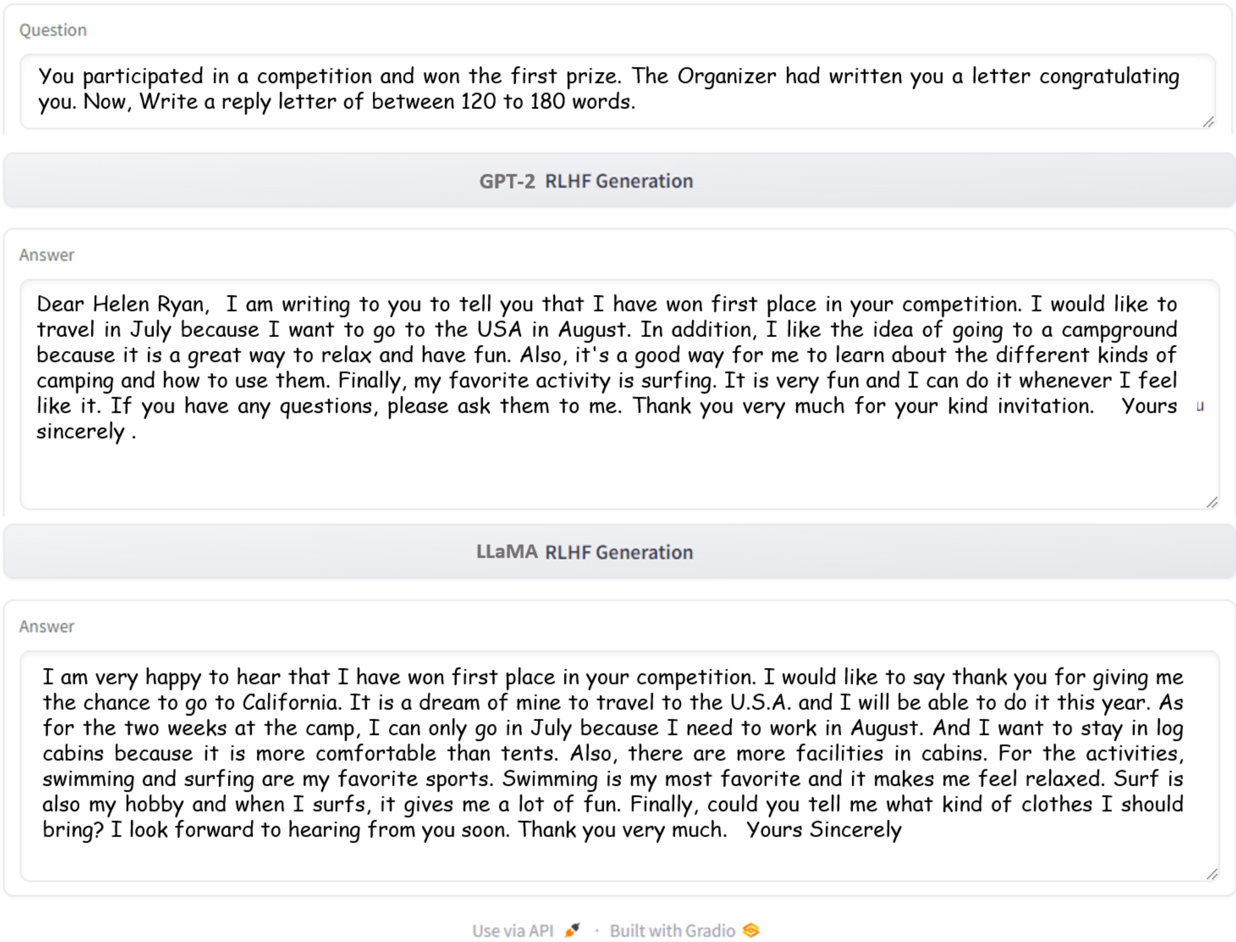


Figure 5.24: Visual Comparison between RLHF Implemented Models

The comparison between GPT-2 RLHF and LLaMA2-7B RLHF responses reveals that both models benefit significantly from reinforcement learning from human feedback (RLHF), but LLaMA2-7B RLHF excels in clarity, coherence, and engagement. The GPT-2 RLHF response, while detailed and relevant, feels slightly disjointed and repetitive. In contrast, the LLaMA2-7B RLHF response offers a highly personalized and smoothly flowing narrative, making it more engaging and readable. RLHF enhances both models by improving the natural flow of text, contextual relevance, and overall human-likeness.

5.5 Assessment of Issues

The development and deployment of advanced AI models like GPT-2 and LLaMA-2 integrated with Reinforcement Learning from Human Feedback (RLHF) bring about various considerations that need to be thoroughly assessed. These considerations encompass societal impact, health and safety concerns, legal and cultural implications, environmental sustainability, and ethical principles. Each of these areas poses unique challenges and opportunities that must be addressed to ensure the responsible and beneficial application. This section aims to provide a detailed evaluation of these issues, highlighting how our work can contribute positively while mitigating potential risks.

5.5.1 Societal Impact

The integration of Reinforcement Learning from Human Feedback (RLHF) in generative question-answering systems like GPT-2 and LLaMA-2 has significant societal implications. These advanced AI systems can greatly enhance user experiences across various applications such as education, customer service, and information retrieval by providing more accurate, relevant, and human-like responses. Improved AI models can democratize access to high-quality information and support personalized learning experiences, thus bridging knowledge gaps and fostering greater inclusivity.

Application in Education:

Our work can be particularly handy in educational settings, providing students with instant, contextually accurate answers to their questions, thereby supporting independent learning and reducing the burden on educators. AI-driven tutoring systems can adapt to individual learning paces and styles, offering tailored feedback and explanations.

Customer Service Enhancement:

In customer service, RLHF implemented LLM models can handle a larger volume of inquiries with high accuracy, improving response times and customer satisfaction. This can free up human agents to deal with more complex issues, thereby optimizing overall service efficiency.

Information Retrieval:

For information retrieval, our enhanced models can go through vast amounts of data to provide concise and accurate responses, aiding professionals in fields such as law, medicine, and research by saving time and improving decision-making processes.

5.5.2 Safety Issues

Developing and implementing AI models involves ensuring that the responses generated are safe and do not cause harm to users. The fine-tuning and RLHF processes requires rigorous testing to prevent the propagation of incorrect or harmful information.

Critical Applications:

In fields like education and emergency response, the reliability of our models will be unquestionable. By integrating human feedback into the training process, we aim to improve the trustworthiness and accuracy of the models' responses.

Enhanced Safety through RLHF:

Our model is safer compared to a standard LLM model due to the use of RLHF, which incorporates human feedback to improve the detection and mitigation of potential risks, ensuring the generated responses are safe and reliable.

5.5.3 Legal and Cultural Issues

Adopting advanced AI systems must adhere to existing legal frameworks and cultural sensitivities. Issues such as intellectual property rights related to the data used for training, compliance with data protection regulations (e.g., GDPR), and the cultural appropriateness of generated content are significant.

Data Protection and Privacy:

Our models need to ensure compliance with data protection regulations, safeguarding user privacy and maintaining the confidentiality of sensitive information.

Cultural Sensitivity:

Incorporating diverse datasets and feedback from various cultural contexts can help our models generate content that is culturally appropriate and respectful, avoiding biases and promoting inclusivity.

5.6 Evaluation of Environment and Sustainability

The training and fine-tuning of large language models like GPT-2 and LLaMA-2 require substantial computational resources, which can have a significant environmental impact due to the high energy consumption of data centers. The carbon footprint associated with this energy use is considerable, raising concerns about the sustainability of developing and deploying advanced AI models.

By integrating RLHF, we can optimize the training process, enhancing model efficiency and reducing the frequency and duration of retraining cycles. This optimization directly translates to lower energy consumption, as the models become more adept at learning from smaller, more targeted datasets, thus requiring fewer computational resources over time.

Enhanced efficiency through RLHF not only improves the performance of AI models but also contributes to environmental sustainability. By minimizing the computational load, the carbon footprint of running extensive AI training sessions is significantly reduced. This is crucial in mitigating the environmental impact of AI development and promoting greener technology practices.

5.7 Ethical Issues

Throughout this project, adherence to ethical principles was paramount. Ensuring the responsible conduct of research involved rigorous data handling practices, avoiding plagiarism, and maintaining integrity in all phases of the project.

Informed Consent:

Obtaining informed consent for any human feedback incorporated in RLHF was essential to ensure ethical transparency.

Transparency and Accountability:

Ensuring that the decision-making processes of our AI models are transparent and accountable helps build trust with the users.

Chapter 6

Conclusion

6.1 Summary

This thesis provides an in-depth analysis of how Reinforcement Learning from Human Feedback (RLHF) can significantly enhance the generative question-answering capabilities of Large Language Models (LLMs) like GPT-2 and LLaMA-2. The research was driven by the necessity to improve the models' abilities to generate responses that are not only contextually relevant but also indistinguishable from human-generated text, thereby increasing their applicability in sectors such as education, customer service, and healthcare, where interaction quality is paramount.

The methodology adopted involves an innovative application of RLHF, where human feedback is integrated into the training process of LLMs. This method diverges from traditional training methods by incorporating evaluations from human to directly influence model training. Feedback from evaluators is used to adjust the reward mechanisms guiding the model, enabling it to learn not just from vast datasets but also from qualitative human insights about the relevance and appropriateness of its responses.

Experimentally, the study meticulously compared the performance of GPT-2 and LLaMA-2 before and after the application of RLHF. The experiments were designed to measure improvements in various dimensions, including the accuracy, fluency, and contextual alignment of the text generated by these models. The simulations were crafted to evaluate the models' performance through several metrics i.e mean reward trend, standard deviation trend and reward values before and after RLHF. The results clearly indicated that RLHF significantly improves the performance of these models across all tested parameters.

Furthermore, the thesis explored the technical and theoretical implications of applying RLHF to LLMs. It discussed the challenges encountered during the integration of human feedback, such as requirement of efficient mechanisms to collect, interpret, and incorporate feedback into the models without introducing noise or biases. The study also focuses on societal and environmental impacts and ethical issues.

In conclusion, the thesis demonstrated that RLHF provides a viable and effective approach for refining the generative capabilities of LLMs, significantly broadening their utility in practical applications. By improving how these models generate human-like, context-aware text, the research contributes to the broader goal of creating AI systems that can seamlessly and intelligently interact with humans across various domains and contexts.

6.2 Future Works

The successful integration of Reinforcement Learning from Human Feedback (RLHF) with GPT-2 and LLaMA-2 models opens numerous avenues for further research and development. The following areas outline potential directions for future work to build on the progress achieved so far:

6.2.1 Expanding the Dataset

One of the primary future directions involves expanding the dataset to include a more diverse range of question-answer pairs. Incorporating varied topics and contexts can help the models generalize better, improving their performance across different domains. This could involve:

i. **Including Multilingual Data:** Training models on multilingual datasets to enhance their ability to understand and generate responses in multiple languages.

ii. **Specialized Domains:** Adding specialized datasets from fields such as medicine, law, and finance to improve domain-specific question answering.

6.2.2 Enhancing the RLHF Process

Further refining the RLHF process can lead to even better model performance. This can include:

i. **Incorporating More Complex Feedback:** Using more detailed feedback mechanisms that go beyond binary good/bad labels to include nuanced ratings on various aspects of the responses.

ii. **Automated Feedback Systems:** Developing automated systems that can simulate human feedback using synthetic data, thereby scaling the feedback process.

6.2.3 Integration of Additional Feedback Mechanisms

Exploring the integration of other types of feedback can provide a more comprehensive training signal:

i. **User Interaction Data:** Leveraging user interaction data, such as click-through rates and time spent on responses, to provide implicit feedback.

ii. **Expert Annotations:** Using expert annotations for critical applications to ensure the highest quality of responses.

6.2.4 Enhancing Interpretability and Transparency

Improving the interpretability and transparency of trained models can build trust and facilitate broader adoption:

i. **Explainable AI:** Incorporating techniques from explainable AI (XAI) to make model decisions more understandable to users.

ii. **Transparent Reporting:** Providing clear documentation and transparent reporting on model development, training processes, and evaluation metrics.

6.2.5 Real-Time Monitoring and Fault Detection

Implementing real-time monitoring and advanced fault detection algorithms can improve the robustness and reliability of AI models in dynamic environments:

i. **Anomaly Detection:** Developing systems to detect anomalies in real-time, ensuring the model's outputs remain accurate and trustworthy.

ii. **Adaptive Learning:** Implementing adaptive learning techniques that allow models to update and improve based on real-time feedback.

6.2.6 Interdisciplinary Research and Collaboration

Fostering interdisciplinary research and collaboration can lead to innovative solutions and advancements:

i. **Cross-Disciplinary Projects:** Engaging in cross-disciplinary projects that combine expertise from AI, cognitive science, ethics, and domain-specific fields to enhance the capabilities and applications of AI models.

ii. **Collaborative Platforms:** Developing collaborative platforms where researchers, practitioners, and users can contribute to and benefit from shared knowledge and resources.

6.2.7 Economic Impact and Field Tests

Investigating the economic impact of deploying these advanced AI systems across various sectors and conducting extensive field tests are crucial for validating their real-world applicability:

i. **Cost-Benefit Analysis:** Performing cost-benefit analyses to understand the financial implications and potential savings of using AI models with RLHF in different industries.

ii. **Pilot Programs:** Launching pilot programs in real-world settings to gather practical insights and refine the models based on user feedback and performance metrics.

References

[1] Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Vol. 1, pp. 4171-4186). Minneapolis, Minnesota: Association for Computational Linguistics.

[2] Radford, A., Narasimhan, K., Salimans, T., & Sutskever, I. (2018). Improving Language Understanding by Generative Pre-Training. OpenAI Blog.

[3] Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., ... & Amodei, D. (2020). Language models are few-shot learners. In NeurIPS.

[4] Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C., Mishkin, P., ... & Leike, J. (2022). Training language models to follow instructions with human feedback. Advances in Neural Information Processing Systems, 35, 27730-27744.

[5] Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., ... & Liu, P. J. (2020). Exploring the limits of transfer learning with a unified text-to-text transformer. The Journal of Machine Learning Research, 21(1), 5485-5551.

[6] Schick, T., & Schütze, H. (2021). It's not just size that matters: Small language models are also few-shot learners. arXiv preprint arXiv:2009.07118.

[7] Zhang, Y., Sun, S., Galley, M., Chen, Y.-C., Brockett, C., Gao, X., ... & Dolan, B. (2020). Dialogpt: Large-scale generative pre-training for conversational response generation. pp. 270-278.

[8] Wei, J., Bosma, M., Zhao, V. Y., Guu, K., Yu, A. W., Lester, B., ... & Le, Q. V. (2022). Finetuned language models are zero-shot learners. In ICLR.

[9] Sutton, R. S., & Barto, A. G. (2018). Reinforcement Learning: An Introduction. MIT Press.

[10] Christiano, P. F., Leike, J., Brown, T., Martic, M., Legg, S., & Amodei, D. (2017). Deep reinforcement learning from human preferences. Advances in Neural Information Processing Systems, 30.

[11] Serban, I. V., Lowe, R., Henderson, P., Charlin, L., & Pineau, J. (2018). A survey of available corpora for building data-driven dialogue systems. arXiv preprint arXiv:1512.05742.

[12] Zoph, B., & Le, Q. V. (2016). Neural architecture search with reinforcement learning. arXiv preprint arXiv:1611.01578.

[13] Dhingra, B., Liu, H., Yang, Z., Cohen, W. W., & Salakhutdinov, R. (2017). Gated-attention readers for text comprehension. arXiv preprint arXiv:1606.01549.

[14] Narang, S., Undersander, E., Zhang, J., Contreras-Ochando, L., Serban, I. V., ... & Pineau, J. (2018). Neural Conversational Model (NCM): A generative pre-trained transformer model for conversational response generation. arXiv preprint arXiv:1706.01427.

[15] Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., ... & Yih, W.-T. (2020). Retrieval-augmented generation for knowledge-intensive NLP tasks. Advances in Neural Information Processing Systems, 33, 9459-9474.

[16] Zhang, J., Balog, K., & Gerani, S. (2019). Knowledge-enhanced contextual word embeddings. In Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval (pp. 221-230).

[17] Gao, Y., Zhang, Y., Galley, M., Chen, Y.-C., Brockett, C., Gao, X., ... & Dolan, B. (2019). Dialogue generation: From imitation learning to inverse reinforcement learning. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP) (pp. 3317-3326).

[18] Wu, Y., & King's College London. (2021). Pre-training and fine-tuning for neural machine translation: A comprehensive review. Neural Networks, 144, 192-204.

[19] Schulman, J., Wolski, F., Dhariwal, P., Radford, A., & Klimov, O. (2017). Proximal policy optimization algorithms. arXiv preprint arXiv:1707.06347.

[20] Ziegler, D. M., Stiennon, N., Wu, J., Brown, T., Radford, A., Amodei, D., ... & Leike, J. (2020). Fine-tuning language models from human preferences. arXiv preprint arXiv:1909.08593.

[21] TechTarget. (2024). Large Language Model (LLM). Retrieved from https://www.techtarget.com/whatis/definition/large-language-model-LLM

[22] Dogoodwithdata. (2024). Large Language Models and Generative AI - Part 1. Medium. Retrieved from https://medium.com/@dogoodwithdata/large-language-models-and-generative-ai-part-1-78f7d801dd2c

[23] DataCamp. (2024). Fine-Tuning Large Language Models. DataCamp. Retrieved from https://www.datacamp.com/tutorial/fine-tuning-large-language-models

[24] Superteams.ai. (2024). Fine-Tuning and Grounding of Large Language Models for Enhanced Performance. Superteams.ai. Retrieved from https://www.superteams.ai/blog/fine-tuning-and-grounding-of-large-language-models-for-enhanced-performance

[25] Hugging Face. (2023). Parameter-Efficient Fine-Tuning using PEFT. Retrieved from https://huggingface.co/blog/peft

[26] Xu, L., Xie, H., Qin, S.-Z. J., Tao, X., & Wang, F. L. (2023). Parameter-Efficient Fine-Tuning Methods for Pretrained Language Models: A Critical Review and Assessment. arXiv. Retrieved from https://arxiv.org/abs/2312.12148

[27] Han, Z., Gao, C., Liu, J., Zhang, J. Q. S., & Zhang, S. Q. (2023). Parameter-Efficient Fine-Tuning for Large Models: A Comprehensive Survey. arXiv. Retrieved from https://arxiv.org/abs/2403.14608

[28] Amanat, M. U. (2023). Fine-Tune Large Language Models with PEFT. Retrieved from https://medium.com/@MUmarAmanat/fine-tune-llm-with-peft-60b2798f1e5f

[29] Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., & Chen, W. (2021). LoRA: Low-Rank Adaptation of Large Language Models. arXiv. Retrieved from https://arxiv.org/abs/2106.09685

[30] Run: AI. (2023). LoRA Fine Tuning. Retrieved from https://www.run.ai/guides/generative-ai/lora-fine-tuning

[31] AWS. (2024). What is Reinforcement Learning from Human Feedback? AWS. Retrieved from https://aws.amazon.com/what-is/reinforcement-learning-from-human-feedback/

[32] Techopedia. (2024). Reinforcement Learning from Human Feedback (RLHF). Techopedia. Retrieved from https://www.techopedia.com/reinforcement-learning-from-human-feedback-rlhf

[33] Hugging Face. (2024). Reinforcement Learning from Human Feedback (RLHF). Hugging Face. Retrieved from https://huggingface.co/blog/rlhf

[34] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is All You Need. Advances in Neural Information Processing Systems, 30, 5998-6008.

[35] Thisislong. (2021). Large Language Models & Transformer Architecture: The Basics. Retrieved from https://medium.com/@thisislong/large-language-models-transformer-architecture-the-basics-2bdd84a6db17

[36] Mvschamanth. (2021). A Brief on GPT-2 and GPT-3 Models. Retrieved from https://mvschamanth.medium.com/a-brief-on-gpt-2-and-gpt-3-models-f4889330328e

[37] Zaiinn. (2021). Autoregressive Models for Natural Language Processing. Retrieved from https://medium.com/@zaiinn440/autoregressive-models-for-natural-language-processing-b95e5f933e1f

[38] Hugging Face. (2024). GPT-2 Model Documentation. Retrieved from https://huggingface.co/docs/transformers/en/model-doc/gpt2

[39] Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M., Lacroix, T., ... & Jegou, H. (2023). LLaMA: Open and Efficient Foundation Language Models. arXiv preprint arXiv:2307.09288.

[40] IBM. (2023). LLaMA-2. Retrieved from https://www.ibm.com/topics/llama-2

[41] DataForest. (2023). LLaMA-2: Generating Human Language with High Coherence. Retrieved from https://dataforest.ai/blog/llama-2-generating-human-language-with-high-coherence

[42] Hugging Face. (2023). LLaMA-2. Retrieved from https://huggingface.co/meta-llama

[43] Spark NLP. (2021). RoBERTa Base Model. Retrieved from https://sparknlp.org/2021/05/20/roberta_base_en.html

[44] Hugging Face. (2020). Pretrained Models. Retrieved from https://huggingface.co/transformers/v2.10.0/pretrained_models.html

[45] GeeksforGeeks. (2021). Overview of RoBERTa Model. Retrieved from https://www.geeksforgeeks.org/overview-of-roberta-model/

[46] Hugging Face. (2023). FacebookAI/roberta-base. Retrieved from https://huggingface.co/FacebookAI/roberta-base

[47] Vtiya. (2021). Dynamic Mask for RoBERTa vs Static Mask for BERT. Retrieved from https://vtiya.medium.com/dynamic-mask-for-roberta-vs-static-mask-for-bert-c997edc9a939

Appendix A

Nomenclature

AI	Artificial Intelligence
NLP	Natural Language Processing
LLM	Large Language Model
PEFT	Parameter-Efficient Fine-Tuning
LoRA	Low-Rank Adaptation of Large Language Models
RLHF	Reinforcement Learning from Human Feedback
GPT-2	Generative Pre-trained Transformer 2
LLaMA-2	Large Language Model Meta AI 2
RAG	Retrieval-Augmented Gen eration
ML	Machine Learning
RL	Reinforcement Learning
PPO	Proximal Policy Optimization
CLC	Cambridge Learner Corpus
FCE	First Certificate in English
RoBERTa	Robustly Optimized BERT Pretraining Approach
BERT	Bidirectional Encoder Representations from Transformers
MLM	Masked Language Modeling
CC-News	CommonCrawl News
NER	Named Entity Recognition
GPU	Graphics Processing Unit
XAI	Explanable AI
QA	Question Answering

Generated using Undegraduate Thesis LaTeX Template, Version 1.0. Department of Electrical and Electronic Engineering, Bangladesh University of Engineering and Technology, Dhaka, Bangladesh.

This thesis was generated on Thursday 21st November, 2024 at 5:01pm.

B.Sc.
Engg.
EEE
BUET

Enhanced Generative Question Answering Using LLMs and RLHF

Usama Ibna Alam
Sudipto Pramanik

June
2024