**CriticalRiver Technologies**

**TextSummarization ChatBot**

**Internship Use Case Document**

**K SRIKAR**

Contents

[1. Introduction 4](#_Toc141699954)

[1.1 Problem statement 4](#_Toc141699955)

[1.2 Objectives and Scope 4](#_Toc141699956)

[1.3 Background on Reinforcement Learning and Language Models\* 4](#_Toc141699957)

[2. Literature Review 5](#_Toc141699958)

[2.1 Key Concepts of Reinforcement Learning and Language Models\* 5](#_Toc141699959)

[3. Data Collection and Preprocessing 6](#_Toc141699961)

[3.1 Description of the Dataset Used 6](#_Toc141699962)

[3.2 Data Preprocessing Steps, if applicable 6](#_Toc141699963)

[4. Model Selection 7](#_Toc141699964)

[4.1 Explanation of the Chosen Model 7](#_Toc141699965)

[4.2 Training the Model, if applicable 7](#_Toc141699966)

[5. Reinforcement Learning with Human Feedback\* 8](#_Toc141699967)

[5.1 Overview of the Approach for RL with Human Feedback 8](#_Toc141699968)

[6. Application Development 9](#_Toc141699970)

[6.1 User Interface (UI) Design and Functionalities 9](#_Toc141699971)

[6.2 Integration of Language Model with the Application 9](#_Toc141699972)

[7. Results and Evaluation 10](#_Toc141699973)

[7.1 Presentation of Application Results 10](#_Toc141699974)

[8. Discussion and Conclusion 11](#_Toc141699976)

[8.1 Key Findings and Insights 11](#_Toc141699977)

[8.2 Limitations and Challenges Faced 11](#_Toc141699978)

[8.3 Conclusion and Future Directions 11](#_Toc141699979)

[9. References 12](#_Toc141699980)

# Introduction

## Problem statement

Develop an AI chatbot to produce concise text summaries from voluminous content, refining its results using real-time user feedback, and ensuring adaptability for diverse content genres, to enhance user experience and information absorption.

## Objectives and Scope

**Objective:** To devise an adaptive chatbot system that leverages the robustness of Large Language Models (LLMs) in generating text summaries, enhanced by reinforcement learning techniques that allow real-time model optimization. By incorporating user feedback as rewards or penalties, the system continually refines its performance, striving to align with human summarization preferences and standards, thus bridging the gap between machine-generated and human-like summaries.

**Scope:** The Reinforcement Learning-based Text Summarization Chatbot leverages the capabilities of Large Language Models (LLM) for the generation of concise text summaries. Designed with a user-centric approach, this system facilitates model refinement by incorporating feedback directly from users. Through an intuitive chatbot interface, users can easily input text and, upon receiving the initial summary, provide corrective feedback. This feedback loop serves as a reinforcement signal, enabling the model to adapt and improve its summarization techniques. While the potential applications of LLMs are vast, this project's primary focus is on refining the art and accuracy of automated text summarization.

## Background on Reinforcement Learning and Language Models\*

**Reinforcement Learning** (RL) is a subfield of machine learning that focuses on developing agents capable of making sequential decisions in a n environment to maximize a cumulative reward. It draws inspiration from behavioral psychology, where agents learn to interact with an environment to achieve desired outcomes through trial and error.

**Key Components in Reinforcement Learning:**

**1. Agent and Environment:** In RL, there are two main components- the agent and the environment. The agent is the entity that learns and takes actions, while the environment is the simulated system with which the agent interacts.

**2. State:** A state represents the current situation of the agent within the environment. The agent’s decisions are based on the state it is in.

**3. Action:** An action is a decision made by the agent in response to the current state of the environment. The agent’s goal is to select actions that lead to desirable outcomes.

**4. Reward:** The reward indicates the immediate benefit or cost associated with the action. It serves as feedback to the agent, informing it about the quality of its chosen action in achieving its objectives.

**5. Policy:** The policy is the strategy or set of rules that the agent uses to select actions based on the current state. The goal is to find the optimal policy that maximizes the cumulative reward.

**6.Value function:** The value function estimates the expected cumulative reward the agent can obtain from a given state while following a specific policy.

**Language Models** form a cornerstone in the domain of natural language processing (NLP), aiming to predict the likelihood of a sequence of words. They are mathematical structures that capture the intricacies, semantics, and grammar of a language, enabling machines to generate and comprehend human-like text.

**Key Components in Language Models:**

**1. Vocabulary and Tokens:** Language models operate on a predefined set of words, known as the vocabulary. Each word or sub-word in this set is termed a token. These tokens encompass the building blocks for any text sequence that the model will encounter or generate.

**2. Context:** The context represents the series of tokens preceding the current token in each sequence. It's essential as LMs rely on this to predict the subsequent token or understand the meaning of a given token within a sequence.

**3.** **Embeddings:** These are dense vector representations of tokens. By capturing semantic and syntactic information, embeddings enable LMs to understand the relationships and nuances between different words.

**4.** **Architecture:** Modern LMs leverage deep learning architectures, most notably the Transformer, which allows them to handle long-range dependencies in text and generate context-sensitive embeddings.

# Literature Review

## Key Concepts of Reinforcement Learning and Language Models\*

**Reinforcement Learning:**

**1. Exploration and Exploitation:** Agents need to strike a balance between exploring new actions to discover their effects (exploration) and exploiting known actions to maximize the reward (exploitation). Effective strategies for exploration enable the agent to discover optimal actions while gradually refining its policy to maximize cumulative rewards.

**2. Reward Function:** Designing an appropriate reward function is crucial as it guides the agent towards the desired outcomes and helps shape its learning process.

**3. Q-Learning:** In RL, Q-Learning algorithm aims to learn the optimal action-value function, which estimates the expected cumulative reward of taking a particular action from a specific state. Q-Learning updates Q-Values iteratively based on the **Bellman equation**, which expresses the Q-Value of a state-action pair as the sum of the immediate reward and the maximum expected future reward achievable from the next state.

- The equation is:

**Q (s, a) = Q (s, a) + α \* [R (s, a) + γ \* max (Q (s', a')) - Q (s, a)]**

where,

**α (Learning Rate):** Controls how much of the new estimate should overwrite the old one. It is a value generally between 0 and 1, typically decreasing over time.

**γ (Discount Factor):** Balances immediate rewards against future rewards. A higher γ values favor long-term rewards, while lower values focus on short-term gains.

**4. Policy Gradient Methods:** These methods directly optimize the policy itself. They’re effective in continuous action spaces and complex environments.

**5. Markov Decision Process (MDP):** An MDP is a formal mathematical framework that models the reinforcement learning problem. It consists of states, actions, transition probabilities, rewards, and a policy.

**Language Models:**

**1. Probability Estimation:** At its core, an LM estimates the probability of a word sequence. Given a sequence of words, it calculates the likelihood of the next word in the sequence.

**2. Tokenization:** The process of converting text into tokens, which can represent words, sub-words, or even characters. This process is essential for preparing textual data for LMs.

**3. Embeddings:** Vector representations of words or tokens that capture semantic meaning. They allow LMs to understand similarity and relationships between different words.

**4. N-gram Models:** An early approach to LMs, where the probability of each word only depends on the last few words. It's defined by *N*, the number of words considered.

**5. Recurrent Neural Networks (RNN):** A type of neural architecture used in earlier LMs that processes sequences by maintaining a memory of previous tokens but struggles with longer sequences.

**6. Transfer Learning:** A technique where a pre-trained model, often trained on a large corpus, is fine-tuned on a smaller, task-specific dataset.

**7. Perplexity:** A metric used to evaluate LMs. It quantifies how well the probability distribution predicted by the model aligns with the actual distribution of the words in the text.

In the **TextSummarization ChatBot** project, Reinforcement Learning (RL) merges with Language Models (LM) to craft a dynamic summarization tool. This union enables the chatbot to produce concise summaries and refine its approach based on user feedback. Leveraging RL, the chatbot tailors summaries through rewards from user interactions, while the integrated LM ensures linguistic accuracy and relevance.

# Data Collection and Preprocessing

## Description of the Dataset Used

The dataset utilized in the "**TextSummarization ChatBot**" project comprises a diverse collection of textual articles sourced from multiple domains such as news articles, research papers, blog posts, and more. Each article in the dataset is paired with a human-generated summary, capturing the core and key details of the original content. This paired structure facilitates the training of the model, allowing it to learn the nuances of effective summarization.

## Data Preprocessing Steps, if applicable

**1. Model & Tokenizer Initialization**: The model and tokenizer are loaded from the **transformers** library using the **MODEL\_CHECKPOINT**, which is set to "t5-small". The T5 model inherently handles many pre-processing tasks internally.

**2. Tokenization for Fine-tuning**: Before fine-tuning the model on feedback data, the input texts and their corresponding corrected summaries are tokenized using the pre-loaded tokenizer. This involves converting the text into a numerical format that can be ingested by the model.

The pre-trained models from the **transformers** library inherently manage many traditional pre-processing steps.

# Model Selection

## Explanation of the Chosen Model

The selection of an appropriate Sequence2SequenceModel is paramount for the success of the “**TextSummarization ChatBot**” project. In this project, we explored a novel approach to text summarization by integrating reinforcement learning (RL) with the state-of-the-art Text-to-Text Transfer Transformer (T5) model.

* T5, developed by Google Research, is a cutting-edge model that unifies various NLP tasks into a text-to-text format
* Built on the foundation of the Transformer model T5 employs both encoder and decoder mechanisms, making the most of the self-attention strategy.

**Reinforcement Learning with Human Feedback** (RLHF) concept is properly integrated to allow the model to learn from user feedback, making the summarization process more aligned with human preferences over time.

This project showcases the potential of combining advanced language models with reinforcement learning for text summarization.

## Training the Model, if applicable

The primary goal of the training process is to refine the T5 model's summarization abilities by iteratively incorporating user feedback through a reinforcement learning (RL) mechanism.

**Initial Setup:**

* We commenced with a pre-trained "t5-small" model sourced from the Hugging Face model hub.
* We constructed an RL environment named **TextSummarizationEnv** with states and actions.

**Training Loop:**

**1. Episode Initialization:** Each episode in the RL setting corresponds to a text summarization task.

2. **Action Selection:** The agent (T5 model) then decides on an action. This decision is governed by an epsilon-greedy strategy. With a probability **epsilon**, the agent explores by randomly choosing an action, while with a probability **1-epsilon**, it exploits its knowledge and always opts to summarize.

3. **Summary Generation and Feedback Collection:** Upon choosing to summarize, the T5 model generates a summary for the input text. This generated summary is then presented to the user for feedback.

The user can:

- **Rate the summary** on a scale (e.g., 1-10).

**-Provide a corrected summary** if they deem the generated one inaccurate or insufficient.

4. **Reward Calculation:** The reward is computed based on how closely the generated summary aligns with the user's expectations. We employ the ROUGE-L metric, a popular choice for evaluating summarization tasks. If a user rating is available, this metric is multiplied by a factor derived from the rating, effectively weighting the reward based on user satisfaction.

5. **Q-table Update:** The Q-table, a fundamental component in Q-learning, stores the expected rewards for each state-action pair. After obtaining the reward, we update the Q-values using the Bellman equation. This ensures that the agent learns and improves its decision-making over time.

6. **Model Fine-tuning:** Every episode concludes with fine-tuning the T5 model based on the collected feedback. The user's corrected summaries serve as new training data.

7. **Epsilon Decay:** To balance exploration and exploitation over time, the epsilon value decays after each episode, causing the model to rely more on its learned knowledge and less on random actions.

8. **Termination:** Training episodes can continue until a specified maximum number of episodes is reached or if the average reward surpasses a predetermined threshold, signalling satisfactory model performance.

**9. Model Saving:** Post-training, the refined T5 model, along with the tokenizer and the Q-table, is saved for future deployment or further refinement.

# Reinforcement Learning with Human Feedback\*

## Overview of the Approach for RL with Human Feedback

The approach for Reinforcement Learning (RL) with Human Feedback involves training an AI agent to make decisions by combining reinforcement learning techniques with human guidance. This approach leverages both the agent’s interaction with the environment and the feedback provided by a human to improve its decision-making process.

**1. Human Feedback Integration**:

Human Feedback is gathered in the form of corrected summary provided by the human users. The feedback serves as valuable guidance to guide the agent’s learning process.

**2. Shaping Rewards**:

The custom defined reward function computes the rewards for the RL agent based on the ROGUE score which is calculated from the deviation between the generated summary and the user-provided summary.

**3. Model Fine-Tuning using Feedback**:

Using the reward signal derived from human feedback the model is fine-tuned. The RL agent (in this case, the T5-small model) adjusts its internal parameters to improve its future summarization predictions, aiming to increase the reward (or decrease the penalty) in subsequent interactions.

**4. Deployment in Real-world Application**:

The Streamlit interface acts as the bridge between the user and the RL-powered summarization agent. It provides a user-friendly platform for users to input text, view generated summaries, and offer feedback, which in turn is used to further refine the underlying model.

The Human Feedback approach merges AI’s learning capabilities with human insights to create adaptable, responsive agents capable of decisions that align with human preferences. This approach sets the stage for collaborative and user-centric AI decision-making.

# Application Development

## User Interface (UI) Design and Functionalities

**Streamlit** is an essential component of the "**TextSummarization ChatBot**" project, providing the backbone for the UI and ensuring a seamless interactive experience for the user.

**UI Design:**

**1. Simple, Intuitive Layout with Streamlit**:

The entire interface of the ChatBot is built using Streamlit, which allows for rapid development of data-driven web applications. The framework’s inherent simplicity and ensures user-centric design.

**2. Interactive Text Areas**:

Streamlit provides widgets, including text areas that form the primary input method for users. The users input their text to be summarized and later provide feedback, all using Streamlit's interactive components.

**3. Action Buttons**:

Buttons like "Summarize", "Fine-tune and Regenerate Summary", and "Continue with a new text" guide users through the process, providing a structured flow to the interaction.

**Functionalities:**

**1. Real-time Summarization**:

The primary functionality of the ChatBot is text summarization. Once users input text and click the "Summarize" button, Streamlit's reactive nature ensures an immediate response, showcasing the summary.

**2. Maintaining Session State**:

One of Streamlit's newer features, the session state, is used to store session-specific data. This means, as users interact with the ChatBot, their texts, summaries, and feedback are maintained across interactions, offering a continuous experience.

**3. Smooth Exit Option**:

Users can end their session simply by typing 'BYE' into the text area. Streamlit's reactive model instantly detects this input and displays a farewell message.

**4. Error Handling and User Feedback**:

Streamlit's ease of use ensures that developers can quickly implement error handling and feedback mechanisms. If users skip steps or input incorrect data, Streamlit can display informative messages to guide them.

## Integration of Language Model with the Application

**1. Model and Tokenizer Initialization**:

The project uses a fine-tuned T5-small model saved at a specified path (**SAVE\_PATH**). Both the model and its tokenizer are loaded using the Transformers library. The pipeline object **summarization\_pipeline** simplifies the process of obtaining summaries, ensuring seamless integration with the Streamlit application.

**2. Fine-tuning the Model**:

The **fine\_tune\_model\_on\_single\_example** function represents the integration of reinforcement learning principles with the language model. When a user provides a "corrected" summary, this function is invoked to fine-tune the model on this new data, allowing the model to adjust and generate improved summaries in subsequent iterations.

**3. Interactive Summarization and Fine-tuning**:

The Streamlit application provides a multi-step interaction with the user:

**Step 1**: The user provides a text to summarize.

**Step 2**: The application generates an initial summary using the **summarization\_pipeline**. The user can then provide a "corrected" or refined summary if they deem it necessary.

**Step 3**: If a corrected summary is provided, the model is fine-tuned using this new data and a refined summary is presented to the user.

This approach ensures that as users interact with the "TextSummarization ChatBot", the system evolves, producing refined and improved summaries over time. The integration emphasizes the importance of human feedback in training models, making AI more aligned with user expectations.

# Results and Evaluation

## Presentation of Application Results

**1. Taking input of the Text to Summarize:**

A screenshot of a chatbot

Description automatically generated

**2. Generating the initial Summary:**

A screenshot of a chatbot

Description automatically generated

**3. Providing Feedback and Regenerating the summary based on the Feedback:**

A screenshot of a chatbot

Description automatically generated

# Discussion and Conclusion

## Key Findings and Insights

**1. Iterative Improvement**: The combination of reinforcement learning, and language models allowed for iterative improvement in summarization. As users provided feedback and corrected summaries, the model became better at generating concise and accurate summaries.

**2. Personalized Experience**: Over time, with continuous feedback from individual users, the model has the potential to offer a more personalized summarization experience, aligning more closely with individual user preferences.

**3. Model Scalability**: Using the T5-small model ensured faster response times while maintaining high-quality summaries. This balance of speed and quality demonstrated the potential for scalability in real-world applications.

## Limitations and Challenges Faced

**1. Misleading Feedback**: The model's performance is reliant on user feedback. If users provide misleading or incorrect feedback, the quality of summaries can degrade.

**2.** **Loss of Nuance**: While the model adapted over time, there were instances where the generated summaries might not have captured all essential points or the nuances. This highlights the challenges in automating text summarization and the importance of continuous learning.

**3. Memory Constraints**: Storing and managing user feedback, especially when handling large volumes of data, can be challenging.

## Conclusion and Future Directions

In conclusion, the TextSummarization ChatBot project represents a significant step forward in the integration of Reinforcement learning with Human Feedback with Language Models. By judiciously incorporating human feedback, this initiative has showcased the potential of training a language model to generate contextually relevant and concise summaries that evolve with user interactions.

The learnings from this endeavor provide a robust foundation for subsequent projects and advancements in the realm of adaptive, AI-driven text summarization.

**Future Directions:**

1. Adapt the model to be proficient in summarizing niche topics or industries such as medical journals, legal documents, or scientific papers.

2. Integrating the bot with Voice Assistants for accessibility, basically making it inclusive for everyone.

# References

* 1. <https://huggingface.co/blog/rlhf>
  2. <https://proceedings.neurips.cc/paper/2020/file/1f89885d556929e98d3ef9b86448f951-Paper.pdf>
  3. <https://openai.com/research/learning-to-summarize-with-human-feedback>
  4. <https://youtu.be/Yo5Hw8aV3vY?si=Z8mpkaSkBjQhSz7p>
  5. <https://youtu.be/HDSNjrxSwqw?si=OgiBP88qmVcM3O-F>