**CriticalRiver Technologies**

**ProjectMangerSimulation**

**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)

[5.2 Collection and Incorporation of Human Feedback into the Language Model 8](#_Toc141699969)

[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)

[7.2 Evaluation of Application Performance and User Feedback 10](#_Toc141699975)

[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 agent capable of learning and making decisions in a grid-based game environment. The

Challenge lies in training the agent to choose the right actions through reinforcement learning, adapting its behavior based on human feedback, and successfully completing predefined correct action sequences for different states in the grid.

## Objectives and Scope

**Objective:** The objective of the “ProjectMangerSimulation” project is to develop an AI agent capable of making informed decisions in a grid-based game environment representing project management scenarios. The AI agent learns to navigate the grid, perform actions, and completer predefined correct action sequences with the guidance of human feedback. Through, reinforcement learning, the agent aims to optimize its decision-making strategies and achieve successful project completion.

**Scope:** The scope of the project encompasses building an AI agent capable of learning project management strategies within a simulated grid environment. The agent’s ability to adapt based on human feedback, follow correct action sequences, and optimize its decision – making process forms the core of this scope. It provides the foundation for exploring the synergy between human expertise and AI capabilities in decision support and learning scenarios.

## 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.

# Literature Review

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

**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.

**6. Language Model:** A statistical model that predicts the likelihood of a sequence of words or characters in a language.

In the project, Reinforcement Learning has been integrated with Language Models to create an AI-driven project management simulation in a grid-based environment. This integration allows the AI project manager to make informed decisions based on learned language rating patterns and RL strategies. The agent’s decisions are influenced by RL principles like rewards and policies, and its ability to communicate is enhanced by language model capabilities.

# Data Collection and Preprocessing

## Description of the Dataset Used

The dataset utilized in the “ProjectManagerSimulation” constitutes a critical component for training and evaluating the AI agent’s decision-making capabilities within a grid-based game environment. The dataset encompasses a diverse range of scenarios, states, actions, rewards, and human evaluations, fostering the agent’s ability to navigate complex tasks effectively.

**Components of the Dataset:**

**1. State Representation:** The dataset encompasses an extensive collection of state representations, each encapsulating the current configuration of the game environment. State information includes Empty state, Ongoing task, Pending task, Resource Shortage, Task completed, Communicate, Progress Review, etc.

**2. Action Sequences:** The dataset records the sequences of actions undertaken by the AI agent in response to its observed states. These actions encompass a variety of choices available within the game environment, such as Initiating tasks, Reallocating resources, Reviewing progress, Adjusting strategies, Defining new Strategies and Communicating changes, etc. Each action is associated with a particular state and contributes to shaping the agent’s overall behavior.

**3. Transition to Next State:** For every action taken by the agent, the dataset documents the resulting transition to the subsequent state in the game environment. This encapsulates the consequences of the agent’s actions, revealing how the game world evolves based on the agent’s decisions.

**4. Reward Signal:** An essential component of the dataset is the reward signal attributed to each interaction. Positive rewards reinforce favorable decisions, while negative rewards discourage suboptimal actions. This reward mechanism plays a pivotal role in training the agent to prioritize actions that lead to positive outcomes.

**5. Human Feedback and Evaluations:** Incorporating the element of human feedback, the dataset includes evaluations and annotations provided by human evaluators for the agent’s decisions. These evaluations could manifest as ratings, labels, or annotations, signifying the quality and appropriateness of the agent’s chosen actions. Human feedback serves as a supervisory signal, guiding the agent’s learning process toward more effective decision-making.

**6.** **Dataset Splitting and Preprocessing:** The dataset is thoughtfully divided into distinct subsets, including a training set and an evaluation set. The training set is utilized to impart knowledge to the AI agent and aids in fine-tuning hyperparameters whereas the evaluation set benchmarks the agent’s performance on unseen data.

Through its incorporation of state representations, action sequences, rewards, human evaluations, the dataset empowers the agent to navigate the grid-based game environment with adaptability and the potential to excel at a variety of tasks.

# Model Selection

## Explanation of the Chosen Model

The selection of an appropriate AI model is paramount for the success of the "ProjectManagerSimulation" project. The Q-learning algorithm has been chosen as the foundation of our project and is underpinned by its compatibility with the project's objectives, which involve enabling an AI agent to learn and make informed decisions within a grid-based game environment.

**Process Flow of the Program:**

**1. Import libraries:** Import the required libraries including `numpy` for numerical operations and `PrettyTable` for tabular visualization of the grid.

**2. Define Correct Action Sequences:** Define a dictionary `correct\_actions\_sequence` that maps states to correct actions that need to be taken to progress through the grid.

**3.** **Define Custom Reward Function:** Create a `custom\_reward` function that calculates the reward based on the desirability change between the current state and the next state. A penalty is applied if the next state is an "Unknown Problem" state.

**4. Define ProjectManagementEnv Class:** Create the `ProjectManagementEnv` class to simulate the grid-based game environment. Initialize the environment with grid size, initial state, and end cell position. The environment has methods for resetting the state, taking actions, and rendering the grid.

**5. Define QLearningAgent Class:** Create the `QLearningAgent` class to implement the Q-learning algorithm. The agent initializes with learning parameters and a Q-table to store Q-values for each state-action pair.

**6. Select Action:** Implement the `select\_action` method that selects actions based on the epsilon-greedy exploration strategy.

**7. Update Q-Values:** Define the `update\_q\_value` method that updates Q-values based on the Q-learning update equation.

**8. Decay Learning Rate:** Implement the `decay\_learning\_rate` method to decay the learning rate over time.

**9. Get User Feedback:** Add the `get\_user\_feedback` method to gather user ratings for the agent's actions.

**10. Initialize Environment and Agent:** Set up the environment with grid size, end cell position, and the maximum number of episodes. Initialize the Q-learning agent with appropriate parameters.

**11. Training Loop:** Run the training loop for a specified number of episodes. Within each episode, reset the state, and perform actions while updating Q-values. Gather user feedback and update the agent's knowledge. Terminate the episode if the agent reaches the end cell or correctly takes actions for each state in the correct sequence.

**12. Evaluate Agent:** Evaluate the agent's performance by running evaluation episodes in the environment. Record the total reward and correct actions count for each episode.

## Training the Model, if applicable

# 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.

The core components are:

**1. Human Feedback Integration:** Human feedback is gathered in the form of explicit ratings provided by human users. The feedback serves as valuable guidance to guide the agent’s learning process.

**2. Shaping Rewards:** Human feedback augments the conventional reward signal that the agent receives from the environment. Positive feedback incentivizes the agent to learn actions that lead to favorable outcomes. Negative feedback discourages the agent from selecting actions that yield undesirable results.

**3. Q-Values Updates with Human Feedback:** The Q-Learning algorithm forms the foundation for incorporating human feedback. Human Feedback updates Q-Values, enabling the agent to learn from its own experiences and human-guided insights.

**4.Handling Noisy Feedback:** Accounting for noisy feedback is crucial due to human subjectivity or uncertainty. RL algorithms consider feedback trends and patterns rather than individual instances to handle variations.

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.

## Collection and Incorporation of Human Feedback into the Language Model

Key points related to the collection and incorporation of Human Feedback into the Language Model of the ‘ProjectManagerSimulation’ project:

**1. Feedback Collection:** Implements a rating system that allows humans to provide feedback on the agent’s actions. Collects the feedback ratings from users after each agent action.

**2. RL Training Loop:** Integrates the human feedback- derived reward function into the reinforcement learning training loop. Allows the AI agent to adapt its behavior over time based on the feedback.

**3. Feedback Analysis:** Analyzes the feedback ratings to understand the user’s perception of the agent’s actions. Evaluates the quality of the actions based on feedback trends and patterns.

These points outline the critical aspects of collecting and incorporating human feedback into the project’s language model, emphasizing its significance in improving the AI agent’s performance and ensuring ethical considerations are met.

# Results and Evaluation

## Presentation of Application Results

A screenshot of a computer

Description automatically generated

-This is a snippet of one of the iterations in the training of the model where the Agent’s chosen action, Current state, reward, and the human feedback loop is being executed with displaying the correct actions count till that point.

A screenshot of a computer

Description automatically generated

-This is a snippet of the end result where the Agent wins. Here, the agent wins doesn’t necessarily mean that the agent got every action correct in every state but it means that it has initially made mistakes and then corrected its actions. This demonstrates the agent’s ability to learn and adjust its behavior based on human feedback.

## Evaluation of Application Performance and User Feedback

**1. Performance Evaluation**

**Evaluation Environment Setup**

**- Grid Size:** The simulation environment was configured with a grid size of 7x7.

**- End Goal:** The agent's objective was to reach the end cell, marked as (6, 6), representing the successful completion of a project.

**Evaluation Episodes**

**- Training Episodes:** During the training phase, the agent learned project management strategies by interacting with the environment. We employed Q-learning as the core reinforcement learning algorithm.

**- Testing Episodes:** The testing phase involved assessing the agent's decision-making capabilities in unfamiliar scenarios. These episodes helped us evaluate how well the agent generalized its learned strategies.

**Performance Metrics**

**- Total Reward:** The cumulative reward obtained by the agent throughout an episode. This metric gauges the agent's ability to make favorable decisions during project management.

**- Correct Actions Count:** The count of correctly chosen actions in alignment with predefined sequences of actions for specific states. This metric assesses the agent's capability to follow best practices.

**2. User Feedback Integration**

**Soliciting User Feedback**

**- User Ratings:** We collected user ratings to gauge the perceived quality of the agent's actions. These ratings provided valuable insights into user satisfaction.

**User Feedback Incorporation**

The collected user ratings were integrated into the Q-learning process. Feedback was utilized to adapt the agent's strategies. Specifically, we considered:

**- Positive Ratings (4 or 5):** The agent retained its current state, as it was deemed satisfactory.

**- Negative Ratings (1 to 3):** The agent transitioned to a new state, encouraging exploration and learning.

**3. Results and Analysis**

**Performance Results**

Our evaluation revealed promising results:

**- Average Total Reward:** The average total reward across evaluation episodes indicated that the agent effectively managed project tasks, with an average score of 11.25.

**- Average Correct Actions Count:** The agent consistently made correct decisions, maintaining an average count of 6.4 correct actions per episode.

**User Feedback Analysis**

User feedback played a crucial role in enhancing the agent's decision-making process. Analyzing user ratings allowed us to fine-tune the agent's actions and align them more closely with user expectations.

# Discussion and Conclusion

## Key Findings and Insights

**1. Enhanced Decision-Making:**

- Finding: Incorporating human feedback into the AI agent's learning process significantly improved its decision-making capabilities.

- Insight: Aligning the agent's actions with human preferences and expertise led to more effective project management strategies.

**2. User Engagement:**

- Finding: Implementing a user-friendly feedback mechanism increased user engagement.

- Insight: A well-designed feedback system encourages users to provide valuable insights, enhancing the agent's learning process.

**3. Future Directions:**

- Finding: There are opportunities for further enhancing the feedback system and agent adaptation.

- Insight: Future plans include exploring advanced feedback mechanisms, adaptive learning techniques, and increased user engagement.

These findings and insights showcase the project's success in utilizing human feedback to enhance AI agent decision-making, improve user engagement, and ensure ethical considerations. It also emphasizes the importance of continuous improvement and a user-centric approach in AI development.

## Limitations and Challenges Faced

**Limitations:**

**1. Limited Feedback Data:** Gathering sufficient high-quality feedback data from human users can be challenging, especially in the early stages of the project.

**2. Bias in Feedback:** Human feedback may be subjective and biased, potentially leading to skewed training data for the AI agent.

**3. Complexity of Real-World Scenarios:** Simulating real-world project management scenarios accurately can be challenging, as these situations often involve intricate, dynamic, and unpredictable factors.

**Challenges:**

**1. Feedback Quality:** Ensuring that the feedback collected is of high quality and truly reflects user preferences and expertise is an ongoing challenge.

**2. Generalization:** Training an AI agent that can generalize its learning from user feedback to new and unseen scenarios is a complex problem.

**3. Dynamic Environments:** Adapting the AI agent to dynamic and rapidly changing project management environments requires ongoing effort.

These limitations and challenges highlight the intricacies of developing a project that incorporates human feedback into AI-driven decision-making.

## Conclusion and Future Directions

**Conclusion:**

In conclusion, the ProjectManagerSimulation project represents a significant step forward in the integration of artificial intelligence, specifically reinforcement learning, into the field of project management. By leveraging human feedback, the project has demonstrated the ability to train an AI agent to make informed decisions and adapt to complex project scenarios.

Through a combination of reinforcement learning algorithms and user feedback, the AI agent has shown promising results in navigating grid-based project management tasks, achieving correct action sequences, and ultimately reaching project completion. This approach has the potential to revolutionize project management by providing real-time, data-driven decision support to project managers and teams.

**Future Directions:**

While this project has made substantial progress, several avenues for future development and improvement are evident:

**1. Enhanced User Interface:** Developing an intuitive and user-friendly interface for project managers to interact with the AI agent will enhance its usability.

**2. Dynamic Environments:** Adapting the AI agent to handle more dynamic project environments with changing objectives and constraints.

**3. Advanced Reinforcement Learning Algorithms:** Exploring more advanced reinforcement learning algorithms, such as deep reinforcement learning, to handle more complex scenarios.

**4. Multimodal Feedback:** Incorporating diverse forms of feedback, including text, voice, and image, to capture richer user input.

**5. Collaborative Learning:** Exploring collaborative reinforcement learning techniques, where multiple AI agents work together or compete to improve their performance.

**6. Industry-Specific Applications:** Tailoring the AI agent to specific industries, such as software development, construction, or healthcare, to address unique project management challenges.

The ProjectManagerSimulation project has laid a solid foundation for the integration of reinforcement learning and human feedback into project management. The future directions outlined above will contribute to the ongoing evolution of this project, making it a valuable asset for project managers across various domains. As technology advances and AI capabilities grow, the potential for optimizing project management processes and decision-making continues to expand.

# References

* 1. <https://realpython.com/python-maze-solver/#step-5-solve-the-maze-using-a-graph-based-approach>
  2. <https://youtu.be/L8ypSXwyBds?si=xCKQnHmd3rk6fZzb>
  3. <https://huggingface.co/blog/rlhf>
  4. <https://www.youtube.com/live/pYpFvmOx9_U?si=4TkfJko8cd0gCy8q>
  5. <https://huyenchip.com/2023/05/02/rlhf.html>