Instructions for *ACL Proceedings

Isaac Peterson
isaac.peterson@usu.edu

Second Author
email@domain

Third Author
email@domain

Abstract

In this project, we explore the application of reinforcement learning (RL) to train an agent capable of navigating a 2D grid environment and identifying geometric shapes with distinct colors. The agent recieves instruction via natural langauge and then the agent moves one space at a time across the grid, using RL techniques to learn an optimal policy for efficiently locating and identifying shapes such as green triangles and red squares. We define the task as a partially observable Markov decision process (POMDP), where the agent's observations are limited to the grid space it occupies. Our approach involves implementing Reinforcement Learning to teach the agent to maximize rewards by minimizing the time and steps required to find and correctly identify a shape. The results demonstrate how reinforcement learning can be applied to shape recognition and navigation tasks, with potential applications in robotics, search-and-rescue missions, and autonomous systems. If time permits, we want to further explore what modifications need to be made in order to command a fleet of agents to accomplish tasks in an optimal manner.

1 Introduction

As technologies continue to advance, the integration of robots into every day life is continuing to increase. As humans rely mostly on speech for communication and instruction, it is essential that robots are also developed to understand and decipher language, executing commands via effectively and efficiently.

There are a myriad of challenges that one confronts when attempting to solve this problem. First and foremost, establishing an interface between the spoken command and correct execution of that command. To properly execute the spoken command, the agent needs to have an accurate understanding of its environment, which in the real world can become extremely complex. Initial attempts were made by utilizing more logic based methods to help the robot understand the task that needs to be solved (Liu, 2016). To mitigate the challenges that come with real world interpretation, many researchers defaulted to simulations to instruct agents in a more structured world.

Video games form a natural challenge for agents, with clear tasks to complete in a very structured world. (Chaplot et al., 2017) used the environment in the video game DOOM™to train their agent to follow natural language instructions, with other researchers using similiary techniques in Minecraft™(Tessler et al.), and even developing their own virtual environments for natural language task completion (Anderson et al., 2017) (Wang et al., 2024).

Our goal is to use the simplified environment in Minigrid (Chevalier-Boisvert et al., 2023) to explore the process of natural langauge task completion, with more research on the multi-agent natural language task completion problem. Where instead of instructing a single agent to complete a task, we instruct a fleet of agents and explore the challenges that come with the increased number of agents along with solution to address these challenges.

2 Related Work

Most of the work in this project will be based on the work done by (Chaplot et al., 2017). They utilize a combination of large language models and reinforcement learning to help an agent understand specific instructions to navigate in the game environment of doom. Others utlized the world of minecraft to help agents complete tasks, with less of a focus on instruction and more on generalized learning rather than interpreting language (Oh et al., 2017), (Tessler et al.). Combining the implementation of (Chaplot et al., 2017) with the Minigrid environment (Chevalier-Boisvert et al., 2023), (Chevalier-Boisvert et al., 2018) we can further explore the challenging problem of robotic instruction with large language models, with the added challenge of addressing multiple agents.

3 Methods

The Minigrid environment is available on a public repository in github. Which will form the foundation for the work that we will present in this research.

References

2016. *Natural-Language-Instructed Industrial Task Execution*, volume Volume 1B: 36th Computers and Information in Engineering Conference of *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*.

Peter Anderson, Qi Wu, Damien Teney, Jake Bruce, Mark Johnson, Niko Sünderhauf, Ian Reid, Stephen Gould, and Anton van den Hengel. 2017. Vision-and-language navigation: Interpreting visually-grounded navigation instructions in real environments.

Devendra Singh Chaplot, Kanthashree Mysore Sathyendra, Rama Kumar Pasumarthi, Dheeraj Rajagopal, and Ruslan Salakhutdinov. 2017. Gated-attention architectures for task-oriented language grounding.

Maxime Chevalier-Boisvert, Dzmitry Bahdanau, Salem Lahlou, Lucas Willems, Chitwan Saharia, Thien Huu Nguyen, and Yoshua Bengio. 2018. Babyai: A platform to study the sample efficiency of grounded language learning. *arXiv preprint arXiv:1810.08272*.

Maxime Chevalier-Boisvert, Bolun Dai, Mark Towers, Rodrigo Perez-Vicente, Lucas Willems, Salem Lahlou, Suman Pal, Pablo Samuel Castro, and Jordan Terry. 2023. Minigrid & miniworld: Modular & customizable reinforcement learning environments for goal-oriented tasks. In *Advances in Neural Information Processing Systems 36, New Orleans, LA, USA*.

Junhyuk Oh, Satinder Singh, Honglak Lee, and Pushmeet Kohli. 2017. Zero-shot task generalization with multi-task deep reinforcement learning.

Chen Tessler, Shahar Givony, Tom Zahavy, Daniel J Mankowitz, and Shie Mannor. A deep hierarchical approach to lifelong learning in minecraft.

Hanqing Wang, Jiahe Chen, Wensi Huang, Qingwei Ben, Tai Wang, Boyu Mi, Tao Huang, Siheng Zhao, Yilun Chen, Sizhe Yang, Peizhou Cao, Wenye Yu, Zichao Ye, Jialun Li, Junfeng Long, Zirui Wang, Huiling Wang, Ying Zhao, Zhongying Tu, Yu Qiao, Dahua Lin, and Jiangmiao Pang. 2024. Grutopia: Dream general robots in a city at scale.

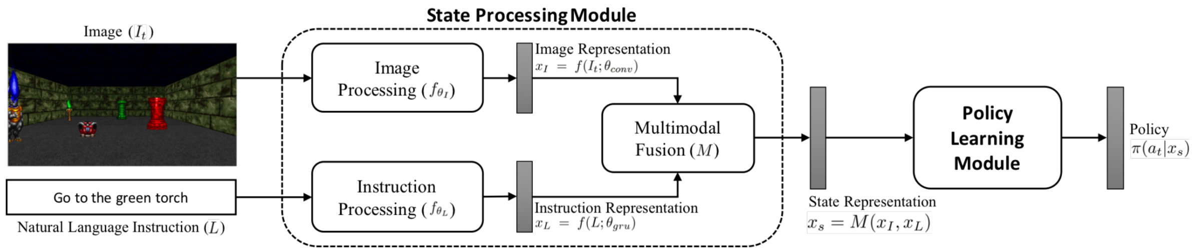

Figure 1: State processing method as developed by (Chaplot et al., 2017).

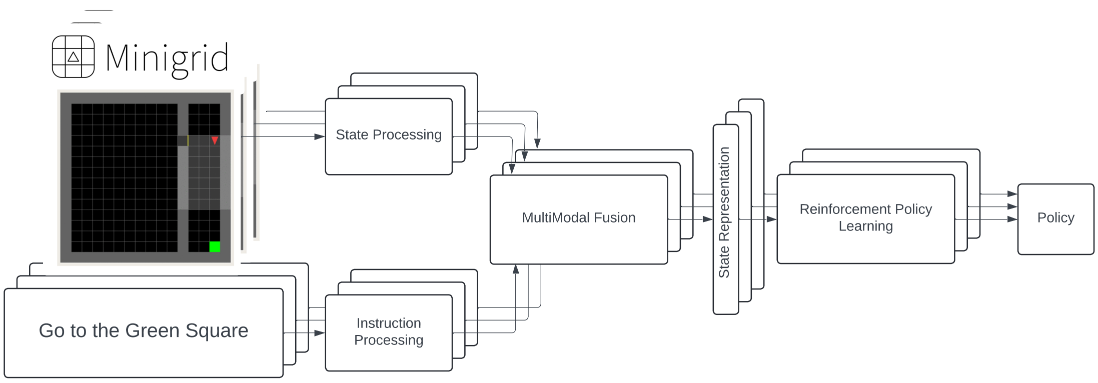

Figure 2: Our approach to solving the multi-agent natural language task-solving problem, showing layers for each agent.