

**College of Engineering**

**COMP 491 – Computer Engineering Design**

**Project Proposal**

**Replication of Beating Atari with Natural Language Guided Reinforcement Learning**

**Fall 2019**

**Participant Information:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **ID** | **Email** | **Phone** |
| Ömerhan Kolukısa | 60346 | okolukisa16@ku.edu.tr | 0 (531) 271 59 98 |
| Engin Deniz Tekalp | 59860 | etekalp16@ku.edu.tr | 0 (538) 683 92 99 |
|  |  |  |  |

**Project Advisor:**

**Abstract**

Multimodal embeddings and Reinforcement Learning (RL) are exciting and fast developing sub-fields of deep learning. We find them very interesting. That’s why we picked as our project to first replicate, then improve on a Stanford project [1]. The project is from the course Natural Language Processing with Deep Learning and it was chosen as the best project in the class of 2017. The project was implemented using PyTorch and our reimplementation will be in Knet. The project involves training an RL agent with natural language augmentation to beat the Atari game Montezuma’s Revenge which is considered the hardest game for RL because of sparse rewards. The main advantage of natural language guidance is to allow intermediate rewards alleviating the sparse rewards problem. Moreover, for the RL agent to understand natural language guidance, multimodal embeddings between commands and game states are utilized. In the end, we aim to first replicate, then surpass the reported performance of [1] measured by game score and training time.

**Table of contents**

[Section 1 Introduction 4](#_Toc348380221)

[1.1 Concept 4](#_Toc348380222)

[1.2 Objectives 4](#_Toc348380223)

[1.3 Background 4](#_Toc348380224)

[Section 2 S/T methodology and associated work plan 4](#_Toc348380225)

[2.1 Methodology 4](#_Toc348380226)

[2.2 Work Package Descriptions 4](#_Toc348380227)

[2.3 Demonstration 5](#_Toc348380228)

[2.4 Impact 5](#_Toc348380229)

[2.5 Risk analysis 5](#_Toc348380230)

[2.6 Gantt Chart 5](#_Toc348380231)

[Section 3 Economical and Ethical Issues 5](#_Toc348380232)

[Section 4 References 5](#_Toc348380233)

1. Introduction
   1. Concept

Our project is replicating [1] which was a Stanford project named Beating Atari With Natural Language Guided Deep Reinforcement Learning. Reinforcement Learning is an approach for training agents to discover the optimal sequence of actions in an environment in which agents can take discrete actions and receive rewards. The project chose Montezuma’s Revenge which is an Atari game, to focus on and they used OpenAI Gym as their programming environment. While Reinforcement Learning algorithms have found great success across many other games, Montezuma’s Revenge is not “solved” due to its challeging nature. By not being solved we are referring to the fact that trained agents fail to get perfect scores. Since rewards are received only after a long sequence of actions, training may not converge as the agents fails to discover optimal sequences of actions. And this problem is referred to as sparse rewards. The project aims to alleviate this problem by giving natural language guidance to the agent. For the agent to understand natural language commands i.e. English, matching commands and game states i.e. sequence of frames need to have matching representations. In the context, of deep reinforcement learning, matching representations imply vector representations in the same embedding space. By giving natural language commands, Kaplan et. al. [1] discovered that they can help alleviate the sparse rewad problem and estabilish communication between humans and reinforcement learning agents via English.

Our project has 2 primary objectives. First of all, we aim to replicate Kaplan et. al.’s results [1]. The original project was implemented using the Pytorch framework and we intend to replicate their results using Knet. The second objective is by trying to solve issues we might discover, using more advanced baseline models and otherwise conducting various experiments, try to improve the reported performance.

Many of the most common Reinforcement Learning algorithms struggle in Montezuma’s Revenge because of the mentioned sparse reward problem. Typically, the state of the art performance is achieved by models that have components that incentivize exploration [7,8]. Our project doesn’t use those additions that incentivize exploration [1] partly because the state of the art in Montezuma’s Revenge has advanced since 2017 when the original project was completed. This could be an opportunity for us to improve on [1], or experiment with whether natural language guidance and exploration are synergystic. The bimodal embeddings between game states and English commands is achieved in 2 steps. First of all, the two modalities are encoded into vectors. Secondly, the similarity of the the vectors are measured by their dot product. And given a dataset of matching gamestates and English commands, the encoders are trained to maximize the similarity of matching representations.

Our methodology can be briefly summarized in 3 parts. The first part will be training a Reinforcement Learning agent on Breakout. Breakout is an easier Atari game and working on it first will allow ourselved to develop our understanding and implementation skills. Next, we will train a Reinforcement Learning agent with natural language guidance on Montezuma’s Revenge. And lastly, we will design and conduct various experiments to try to improve on the reported performance of [1].

Our expected outcomes are an agent that can play Montezuma’s Revenge well, that was trained via Reinforcement Learning and that can act on natural language guidance i.e. English commands.

* 1. Objectives

There are 2 objectives.

The first objective is to replicate [1]. This project was implemented in Python using Pytorch and the replication will be in Julia using Knet. The output should be a Reinforcement Learning Agent that can be guided through natural language to achieve performance observed in [1] for the Atari game Montezuma’s Revenge. We aim to complete this objective by the 22nd of November.

The second objective is to experiment with the further study ideas in [1] as well as other methods we will use to remedy potential shortcomings we will identify as we reimplement the project. The output should be a Reinforcement Learning Agent that can be guided through language that has superior performance compared to [1]. We aim to continue such experiements until the end of the semester.

* 1. Background

The problem to be adressed is composed of three parts. The first part is how can natural language and the states of some environment such as the atari game Montezuma’s Revenge be mapped into the same embedding space. The second part of the problem is, how can RL be used to train agents to play Montezuma’s Revenge which is a tough environment for RL agents due to sparse rewards. The third part of the problem is how can we make use of natural language guidance to improve the performance of reinforcement learning algorithms.

All 3 parts are important problems. First of all multimodal embeddings are important, because being able to represent knowledge across different modalities can improve the accuracy of representation. Moreover, it enables different multi modal applications such as automatic image captioning. Secondly, reinforcement learning in environments with sparse rewards is important because it enables the training of agents that need to complete action sequences over a long time period and perform exploration. Thirdly, natural language guidance is important because it is an example application of human machine interaction. Furthermore, this project is timely because both reinforcement learning and multimodal embeddings are fast developing and important research areas that attract a lot of attention. In fact, from an arxiv search of papers with the key word “modal embedding” generates 123 papers published in the last 12 months [2]. Likewise, an arxiv search of papers with the key word “reinforcement learning” generates 2512 papers published in the last 12 months. The high amount of papers published in these areas demonstrates the attention these problems have been receiving and therefore the timeliness of our project topics.

As mentioned the problem consists of 3 parts. However, the third part of the problem is a toy problem proposed by [1]. Therefore, there is not much literature on using Natural Language guidance to beat Atari.

There are 2 aspects to multi-modal embedding. First is the encoder networks necessary for a dense representation of the 2 modalities. Then there is a need for some similarity measure based on the vector outputs of the encoders. Intrinsic similarity measures i.e. comparing the similarity of two vectors, rely on a human’s labeling of two things being similar. So, they are inherently subjective and thus, it is hard to estabilish state of the art performance given that labels could be rather arbitrary. However, a popular similarity metric is cosine similarity as mentioned by Goldberg [4]. The paper our project is based on used the the sign of the dot product of the two vectors as a similarity measure [1], where positive dot product meant the two things are similar.

For encoder architecture choices, one recently published paper by Laina et. al. [5] used Gated Recurrent Units(GRUs) for feature extraction from text and Convolutional Neural Networks (CNN) for feature extraction from images. The paper our project is based on [1] used Long-Short Term Memory Networks (LSTMs) for feature extraction from text and CNNs for feature extraction from image. Whereas most projects use CNNs for image feature extraction, there is a variety of archictectures that are used for text feature extraction. However, since our embeddings will be limited to a small vocabulary given that there will be limited amount of commands, the choice of architecture for textual feature extraction may not be as critical.

As for reinforcement learning algorithms training on Montezuma’s revenge the state of the art is tracked by a github page [6]. Of the reinforcement learning algorithms trained without human supervision data, the best performance, defined by the highest score attained, is achieved by Burda et. al. [7]. Their approach introduces a random network distillation bonus which incentivizes exploration with cheap computational overhead. Since Montezuma’s Revenge is a game with sparse rewards, it makes sense that approaches that reward exploration improve performance. Compared to the 10000 score achieved by [7], Bellemare et. al. Achieved a score of 6600 [8]. Bellamare et. al. Proposed a method for modeling uncertainty and rewarding exploration of areas with greater uncertainty [8].

1. S/T methodology and associated work plan
   1. Methodology

To achieve the first objective i.e. to replicate the results of [1] on Montezuma’s Revenge we will first familiarize ourselves with the gym environment and various Reinforcement Learning (RL) algorithms (e.g. Asynchronous Advantage Actor-Critic). To do that we will use a simpler Atari game called Breakout and we will train A3C for Breakout. This approach is similar to [1].

After familiarizing ourselves with RL and OpenAI Gym, we will implement some subtask rewards to gain experience working with such rewards. These rewards can be penalizing loss of life and rewarding certain paddle behaviours on Breakout. Implementing subtask rewards will be a necessary component of reimplementing [1].

Next, we will train bimodal embeddings and utilize them for subtasks on Breakout, again in a similar progression to [1].

Once we have gained experience training RL agents, working with subtask rewards and bimodal embeddings we will have developed our understanding of RL and ability to implement the components of the model in [1]. So, we will be ready to move on to Montezuma’s Revenge which is the target task.

To complete the first objective we will have to train a model that can perform as well as the agent in [1] on Montezuma’s Revenge. To do that we will have to implement the subcomponents mentioned previously. More specifically, we will have to reimplement A3C, introduce subtask rewards to the training of A3C and then introduce bimodal embeddings to the subtask reward system.

Our approach to completing our first objective is similar to how [1] have developed their model. As such it will not be a novel approach. However, because we will be replicating the results in [1], replicating their progression will allow us to effectively develop our understanding of the relevant concepts.

Once we have replicated the results in [1], we will move on to our second objective where we will try to improve on [1]. It is difficult to specify how we can improve on the results beforehand. However, we can try a diferrent baseline model, introduce intrinsic motivation as mentioned in [1], try to generate more training data for bimodal embeddings, improve the bimodal embedding procedure and try different natural language guidance. More likely however, we will come up with specific and different ideas as we develop our understanding of the relevant concepts. Improving on [1] will be a novel task, however, we will have to conduct several experiments some of which will inevitably fail. So, experimenting may not necessarily be an effective process.

* 1. Work Package Descriptions

Detailed work package descriptions with contributors, objectives, task descriptions, deliverables and milestones. A template is given below.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Work package number | **1** | Start date or starting event: | | | **Week 1** | |
| Work package title | **Background research, etc......** | | | | | |
| Participant number | 1 | 2 | 3 | 4 | | 5 |
| Participant name |  |  |  |  | |  |
| Weeks per participant |  |  |  |  | |  |
| **Objectives**   * List objectives of this WP..... | | | | | | |
| **Description of work**  List a task list with start-end weeks.....  **T1.1** (w1-w12) Maintain project web site  Establish and maintain a project web site for providing internal and external access to appropriate project related information.  **T1.2** (w1-w2) Task 2  . | | | | | | |
| **Deliverables**  **D1.1** (w3) Plan for the project web site.  **D1.2.x**  .  .  . | | | | | | |
| **Milestones**  **M6.1** (w3,w5,…,w11) Bi-weekly website updates  **M6.2**  .  . | | | | | | |

* 1. Demonstration

There will be 2 components of performance. First, the amount of frames or training time necessary for convergene and secondly scores first on Breakout, then on Montezuma’s revenge.

In other words, models that can train faster and achieve higher scores will be superior. Our first objective was to replicate the results of [1] by the 22nd of November. So, our performance on our first objetive will be measured by the score and training time necessary to replicate the results of [1].

Our second objective was to improve on [1] until the end of the project. So by the final demonstration we hope to achieve superior performance. More specifically, compared to [1] our model to should train faster and/or achieve a higher score.

In terms of the actual final demonstration, since our project is training an agent that can play Atari games using RL with natural language guidance, there is an obvious demonstration. We can show the agent playing Montezuma’s revenge, with natural language comments visible in the screen.

* 1. Impact

There has been great progress in the field of Deep Reinforcement Learning. However, performance in environments with sparse rewards that necessitate more exploration like Montezuma’s Revenge lack behind in progress. Whereas a game with dense rewards like Breakout is “solved”, Montezuma’s Revenge is not “solved”. Therefore improving on approached to games like Montezuma’s Revenge is a critical issue in RL. Additionally, representing the states of a world and natural language in the same embedding space or more generally multimodal embeddings are an increasingly important way of representing knowledge.

Our project aims to understand, replicate and improve upon another project [1] which used PyTorch as a framework in its implementation and we will try to replicate the results of [1] using Knet. By replication of [1] we aim to develop our understanding of RL and multimodal embeddings. Moreover, when we try to improve on [1], we hope to contribute to the understanding how RL and natural language guidance can be used synergystically. If we can make any improvements, this would manifest itself as faster convergence and/or higher scores.

* 1. Risk analysis

Training our models will take as much as 12-24 hours. This will slow down debugging and our ability to iterate on our models. We may counter this problem by using cloud services such as AWS. We can run multiple experiments simultaneuosly by asking for multiple GPUs and by using both of our accounts.

Another risk we may have is the possibility of bugs in the code given by [1]. We do not know beforehand any issues we might encounter while using their code. While reimplementation the project on Knet, it is essential to have the original version of the code. In case there are bugs, we have to spend time debugging or implement our own version which would reduce our speed.

Lastly, after finishing replication we might not be able to substantially improve the project. Although we expect to make improvements by leveraging research since [1] was published and pursuing further research ideas in [1]. Still, experiments will be time consuming and their outcomes will be uncertain. We will mitigate this risk by being more ambitious with our reimplementation schedule. By leaving more time for experiments, we can create more opportunities for further experiments for ourselves.

* 1. Gantt Chart

Project schedule showing work packages and deliverables throughout the weeks of the semester.

You can prepare your Project Gantt Chart using MS Project (<http://www.matchware.com/en/special/gantt-chart.php>) or similar tools.

1. Economical and Ethical Issues

A significant constraint is the computation time and GPU requirements of training Deep Learning and RL algorithms. We can solve this issue by using cloud services such as AWS which creates an economical constraint. We will have to use the student credit AWS gives us and stay within that budget. Otherwise we will have problems training the RL agent since it can take upto 12-24 hours.

Since engineering is a vital component of modern society, engineers have a moral duty to the public, to their clients, employers and their profession. Their ethical duties include being truthful, avoiding plagiarism, keeping the welfare of the society in mind and being objective. Since we will be reimplementing [1] and we won’t be producing any apps, our project won’t have societal impacts. However, in accordance with the honesty and integrity requirements we need to make clear our project is based on [1] and for improvements we need to cite any other project we might use.

1. References

Give the list of references. All the references should be cited within the text of the proposal report.

1. stanford report

2. <https://arxiv.org/search/advanced?advanced=&terms-0-operator=AND&terms-0-term=modal+embedding&terms-0-field=all&classification-physics_archives=all&classification-include_cross_list=include&date-filter_by=past_12&date-year=&date-from_date=&date-to_date=&date-date_type=submitted_date&abstracts=show&size=50&order=-announced_date_first>

3.

<https://arxiv.org/search/advanced?advanced=&terms-0-operator=AND&terms-0-term=reinforcement+learning&terms-0-field=all&classification-physics_archives=all&classification-include_cross_list=include&date-filter_by=past_12&date-year=&date-from_date=&date-to_date=&date-date_type=submitted_date&abstracts=show&size=50&order=-announced_date_first>

4. goldberg book

5.<https://arxiv.org/pdf/1908.09317.pdf>

6.<https://github.com/reinforcement-learning-kr/rl-montezuma>

7.<https://arxiv.org/abs/1810.12894>

8. https://arxiv.org/pdf/1606.01868.pdf