Interactive Question Answering

Proposal

Gregory Furman
FRMGRE001@myuct.ac.za

Roy Cohen
CHNROY002@myuct.ac.za

Edan Toledo
TLDEDA001@myuct.ac.za

1 Project Description

A primary focus of machine reading comprehension (MRC) research centers around the retrieval of *declarative knowledge* - that is, explicitly stated or static descriptions of entities within a knowledge-base (KB) [38]. Evidence suggests that current MRC and neural methods struggle to fully engage in actual *comprehension* as well as fail to capture an *understanding* of a required question-answering task [34, 38, 40].

The majority of current MRC datasets effectively train models to perform simple pattern matching of words & phrases when attempting to query a knowledge source [39]. This fails to mimic the manner in which humans use interaction and observation in order to gain knowledge about an environment. Furthermore, this simple pattern recognition does not reward information-seeking behaviour that is necessary to answer many natural language questions [22]. On the contrary, procedural knowledge places emphasis on the sequence of actions taken to perform a task as opposed to just a task's completion. Such methods aim to reward the gathering of *declarative knowledge* needed to answer a question.

To this end, we propose interactive question answering (IQA) as a viable solution to the lack of *comprehension* skills current methods fail to develop. IQA sees an agent be tasked with answering questions that require interacting with some dynamic environment [15]. This environment can be in the form of a KB [13], some entity-relation schema, or, more recently, a text-based game [11, 38]. Text-based games have risen in popularity by allowing for language-learning problems to be approached using reinforcement learning (RL) methods in dialogue-like environments [20, 30].

One of the most popular frameworks for text-based games is Microsoft *TextWorld* [11, 30]. TextWorld is an open-source simulator that aims to train RL agents to acquire skills such as decision making and language comprehension. Moreover, it provides a framework for interactive text-based games to be developed in tandem with question-answer pairs.

Using TextWorld, Yuan et al. developed the Question Answering with Interactive Text (QAit) [38] task. Here an agent interacts with a partially observable text-based environment in order to gather information such as the attributes, locations, and existence of objects to answer questions. The QAit task aims to create interactive agents that seek out information and answer questions in a *procedural knowledge setting*.

This project aims to further the state-of-the-art work done by Yuan et al. [39] in the QAit task and improve upon their results by means of implementing an alternative curiosity-driven approach for the RL agent, integrating in some predetermined *contextual* understanding of the environment via a knowledge-graph, as well as attempting to reframe the current QAit task into a sequence modelling problem.

2 Research Motivation

2.1 Problem Statement

The current state-of-the-art (SOTA) model for RL based IQA proposed by Yuan et al. [39] has established a good baseline for future work. However, the model performance fails to achieve significant results above random baselines. Baseline models are also seen failing to generalise to unseen environments. This problem will be approached in the following ways discussed in sections 2.2, 2.3, and 2.4.

2.2 Instilling Curiosity & Stochasticity into Text-Based Agents

Although reinforcement learning has seen success in achieving super-human performance in extremely high dimensional and challenging domains [27], this has all been done in consistent and trained upon environments. Classical reinforcement learning algorithms suffer when attempting to leverage this pretrained knowledge to unseen environments [10]. In addition to this, these algorithms struggle to perform in sparsely rewarded environments that potentially require long sequences of actions before a reward is given [36, 39], such as in the IQA task. The current reinforcement learning approaches in IQA show the performance of using value-based methods such as DQNs [25] and DDQNs [36] with epsilon-greedy policies (which is a greedy algorithm with a small chance of performing a random action). Prior work [36] has shown the extreme difficulty of these methods achieving high rewards in sparse environments due to the use of inefficient random exploration. In addition to this, research [14] has shown that DQN methods are poor at generalizing even in environments with very similar underlying dynamics. To overcome these limitations we propose that using a curiosity-based approach alongside an actor-critic framework will greatly improve performance. Curiosity driven reinforcement learning [7, 23, 29] has shown promising results in increasing the performance of agents in sparsely rewarded scenarios as well as it's ability to generalize to similar but unseen environments. The addition of curiosity to an agent can instill more human-like exploratory behaviour and show the benefit of exploration in partially observable environments. Furthermore the use of an actor-critic stochastic policy, instead of an epsilon-greedy policy, is also believed to aid an agent's ability to explore and adapt better in these unseen environments by allowing agents to learn more robust and general policies [28]. It is also hypothesized that the policy in the IQA task will be simpler to learn and approximate compared to approximating the true value function thereby allowing faster learning and yield a superior policy [33, 35].

2.3 Graph-Based Approach

Providing an RL agent with some contextual understanding about the environment with which it inhabits has been shown to drastically aid performance [3]. This additional context results in action-sequences that are better suited to goal-achievement by providing circumstantial information regarding the action-space. Within the context of TextWorld, many agents fail to generalise and capture necessary meaning and relationships between entities found in the environment [39]. A model failing to develop some contextual-awareness of the world can be detrimental to performance. Current works indicate knowledge graphs (KGs) as useful for expressing supplementary information about the world in order to better facilitate decision making whilst adhering to partial observability [2]. Thus, there is ample motivation for the investigation of equipping an agent with some additional knowledge in order to supplement the learning & decision-making process in a text-based QA system.

2.4 Sequence Modelling Approach

Reframing the QAit task as sequence modelling problem akin to an offline RL problem, whereby optimal policies must be derived from sub-optimal data, gives way for research into sample efficient methods to achieve performant systems. [8] has recently shown the potential for using transformer sequence-to-sequence models as a way to optimize action trajectories through an environment whereby only sub-optimal data was available to learn from. A transformer based approach to action sequence generation can avoid many limitations and difficulties faced by traditional RL methods such as credit assignment. This is especially relevant in the QAit task due to the sparsely rewarded environment structure. Additionally transformers have illustrated strong generalization performance in classical natural language tasks [6]. Lastly, framing the action sequence generation in this manner would allow taking full advantage of existing large models such as GPT or BERT to improve training time and performance.

3 Research Objectives

3.1 Instilling Curiosity and Stochasticity into Text-Based Agents

Value-based RL methods learn some value function that maps an agent's states & actions to some expected cumulative reward from a given state onward [35]. These methods have been observed failing to generalise to out-of-training problems [14, 39]. To this end, literature has indicated that sparsely rewarded environments see this strategy outperformed by *curiosity* driven approaches as well as an *actor-critic* framework [7, 29]. Curiosity driven methods aim to encourage exploration by rewarding actions that lead to unseen states. The actor-critic framework enables teaching an agent to act directly upon an environment while also approximating some value function to help further guide decision-making. By applying the aforementioned methods, we aim to improve upon Yuan et al's findings and achieve better generalisability to unseen environments. We also aim to outperform existing baseline models developed by the QAit task [39].

3.2 Graph-Based Approach

Given the nature of the partially observable environment with which the RL agent inhabits, reasoning about the true underlying world-state is challenging. Furthermore, comprehending environmental complexities via potentially incomplete or inaccurate text-based descriptions alone fail to act as sufficient representations of this broader, unobservable world. Thus, KGs can express supplementary information about the world in order to better facilitate decision making whilst adhering to partial observability. Moreover, this information about the environment in the form of KGs can be provided to the agent in a manner that encourages the development of contextual understanding [1].

Therefore, by embedding specific details about an environment into a knowledge-graph and allowing an agent to utilise it to inform decision making, we posit that improvements on state-of-the-art accuracy on the QAit task can be achieved [39].

3.3 Sequence Modelling Approach

By converting the current baseline SOTA architecture into a single sequence-to-sequence transformer, we hope to take advantage of the zero-shot generalisation performance and scalability of transformers [6] to improve upon training time, performance, and sample efficiency in the QAit task. We also aim to leverage advances in current language modelling such as BERT [12] to inject "weak common sense" into the model.

4 Related Work

4.1 Curiosity

For many sparsely rewarded environments, shaping the reward function is not possible. Using random exploration methods rely upon the agent stumbling onto the goal state by chance. This can be practically impossible in large environments and will result in failure to learn. Curiosity or intrinsic motivation can be seen as a new way of learning which requires no extrinsic rewards from the environment. It is a controlled form of exploration. The two most popular formulations of intrinsic reward can be grouped as follows: The first class of methods try to encourage the agent to explore states it hasn't seen before. [5] have shown great results in very sparsely rewarded environments, such as Montezuma's Revenge, using such exploration with DQN. The second class of methods focus on encouraging the agent to take actions that lower the error in the agent's ability to predict the consequences of its actions. Intuitively this aims to increase the agent's knowledge about the environment [9, 31, 32]. Measuring the novelty i.e. how different a state is from a previous state, or building an internal environmental model to predict the next state can be a difficult problem in high dimensional state spaces. This is compounded with environment stochasticity and noise which ultimately makes intrinsic reward calculation difficult.

4.1.1 Intrinsic Curiosity Module. [29] have derived a method, belonging to the second class, to avoid these difficulties by using a model that only predicts the environmental changes that are caused because of the agent's actions or those that affect the agent. It works by transforming the agent's observation into a feature space where only the relevant information is represented. The agent itself learns the feature space using a neural network trained on an inverse task of predicting the agent's action given the current and successive state. By training on the inverse task, the model's features embedding space only concerns itself with factors that affect the agent. The feature space of the inverse model is then used to train a new neural network to predict the next state's feature representation given the current state's features and action. The agent is then supplied with the prediction error as an intrinsic reward. Refer to figure 1 for an illustration. The intrinsic reward is summed with the extrinsic reward, such as $r_t = r_t^e + \beta r_t^i$ where β balances the exploration and exploitation, to ultimately solve the task given. [29] shows how curiosity greatly improves exploration as well as acts as a mechanism for agents to learn skills that potentially help in future scenarios. This exploration helps in solving very sparse reward settings. [29] also shows that fine-tuning an agent that has been trained with curiosity and extrinsic rewards is more beneficial (learns faster and achieves higher scores) than training from scratch for new unseen environments. This shows how curiosity helps agents generalise to new environments. This method however could fail in learning a sufficient feature representation of the environment thereby leading to poor performance.

4.2 Transformers

Traditional Recurrent Neural Network models have the drawback of losing information in extremely long sequences. Even though LSTMs can retain some memory, due to the sequential structure, the output is not directly influenced by each sequence element and long term dependencies can still be forgotten. Attention mechanisms were created to allow each item in the sequence some level of direct influence over the output, ultimately to give a model more flexibility in its memory capability. Transformers [37] drop the recurrent structure for a purely attention based architecture. Transformers make use of a stack of 6 encoder and 6 decoder modules. Each encoder contains two sub layers: a self attention layer and a linear feed forward network. Each decoder contains three sub layers: a self attention layer, an additional attention layer used over the encoder outputs and a linear feed forward network. Since transformers have no recurrence, meaning that all inputs are given in no special order or position, problems that require temporal information such as language modelling create positional encodings that are added to the model to give additional sequential information about inputs. The non-sequential nature of transformers allow for significant training speed up via utilising parallelism.

4.3 Previous Approaches to Text Games

In 2015, Narasimhan et al. introduced LSTM-DQN: a two-part architecture for deep-reinforcement learning consisting of an LSTM for the representation of textual input and a DQN to jointly learn state representation and action policies [26]. This was tested on two text-based games both derived from the *Evennia* framework and was found to outperform baseline models [26, 30].

That same year Deep Reinforcement Relevance Network (DRRN) was proposed as a means of playing text-based games [19]. This saw an architecture that uses separate embedding vectors for the representation of the action-state space. These are then combined with some interaction function to approximate the Q-function i.e. value function. This too was evaluated on two text-based environments and gained state-of-the-art performance.

4.3.1 Pretraining. Given the extremely sparse nature of language learning in text-based games, methods have been proposed to reduce this so-called *unbounded* action-space [30].

Current literature indicates that the integration of pre-training certain aspects of a model provide significant performance improvements over other methods [2, 18]. For an agent to successfully manage to generalise to an unseen environment, transfer learning can be used to help agents better perform by leveraging some pre-determined external KB. Similarly, pretrained models are able to significantly improve the results of agents playing text-based games, as shown by [17]. In 2019, Ammanabrolu & Riedl show how specific parts of their KG DQN can be pre-trained and utilise existing question-answering methods. This showed to improve the model's convergence and provides a way for knowledge to be transferred between models for text-based games [2].

5 Procedures and Methods

5.1 Initial Code Base

As the simulator developed by [39] is publicly available on GitHub, with an MIT license indicating limitless extendability, our initial task will be to set up the already existing code base. After installing necessary dependencies required for running, we will have access to the baseline platforms, data sets, and data necessary for evaluation of our problem statement. Moreover, this code base will be utilised by the procedures outlined in Sections 5.2, 5.3, & 5.4. The initial code base which will be altered for each approach is described as follows:

5.1.1 Environment & Difficulty . Following work done by Yuan et al., Microsoft Textworld will be used in conjunction with the QAit [39] world & question pair generator. This will be done in order to generate sets of question-answer pairs related to TextWorld environments. QAit aims to test an agent's language comprehension abilities using tasks that require an understanding of locality, existence, and attributes. All environments are procedurally generated by sampling from the world setting distribution (see Table 1), where environment-configurations are distinguished into Fixed Map and Random Map categories. Fixed Map generates text worlds with a fixed number and layout of location-names, rooms, and random objects within rooms. Alternatively, the Random Map setting sees the generation of text worlds with a uniformly-random layout of rooms, names, and objects.

Yuan et al.'s methodology for evaluating agent performance will be used. This sees average QA training-accuracy being calculated over five training sets. These will see Fixed and Random Map Types having Number of Games settings of 1, 2, 10, 100, 500, and *unlimited*, respectively. In the context of our experiments, unlimited games will see games being generated during training with environment-parameters (such as location, number of objects, map type, and random seed) being randomly sampled. This process of randomly generating environments is carried out to evaluate the generalisation capabilities of the agent.

An agent's accuracy will be measured using zero-shot evaluation. This constitutes 500 never-before-seen games being held out during training with each containing a single question needing to be answered. Yuan et al. proposed this as a means of assessing a model's generalisation abilities in a manner akin to the test set of supervised learning problems.

5.1.2 Question Types. There are three types of questions that the agent will be attempting to answer based on these generated worlds. First is the location type question such as "Where is the can of soda?", where a suitable answer would be "fridge". Second is existence type questions such as "is there a raw egg in the world?" where the answer would simply be yes or no. The last type of question which is the most difficult is attribute type questions such as "is apple edible" where the answer is also yes or no. Objects within the attribute question setting are given arbitrary and randomly made-up words to discourage agent memorization of values such as an apple always being edible.

	Fixed Map	Random Map
# Locations, N_r	6	$N_r \sim Uniform(2, 12)$
# Entities, N_e	$N_e \sim Uniform(3 \cdot N_r, 6 \cdot N_r)$	
Actions / Game	17	17
Modifiers / Game	18.5	17.7
Objects / Game	26.7	27.5
# Obs. Tokens	93.1	89.7

Table 1. Statistics of the QAit dataset. Numbers are averaged over 10,000 randomly sampled games. [39]

5.1.3 Baseline Reinforcement Learning Agents. Within these procedurally generated text environments, baseline reinforcement learning agents that make use of Deep Q Networks [25] will be used with an epsilon greedy policy. Multiple Transformer [37] models will be used to encode the textual observation along with other relevant environment information into contextual vector representations that are used by agents to decide upon the action to take. During training time, the admissible actions are available for agents to use but at evaluation time, actions must be constructed from the vocabulary in the simple form of a triple (Action, Modifier, Object). The question answering model functions differently depending on the question. For location type questions, each token in the final state observation string is used to construct a probability distribution of potential answers. Here, the highest probability token is selected as the answer. For attribute and existence questions, the probability distribution is simply over the two possible answers: yes and no.

5.2 Instilling Curiosity and Stochasticity into Text-Based Agents

5.2.1 Curiosity. For instilling curiosity into agents, we will be adapting an existing approach by introducing an intrinsic curiosity module (ICM) [29] (see section 4.1.1). This module aims to learn relevant features about the agent's text observations as well as use these features to predict the next state. The module then uses the scaled prediction error as a form of intrinsic reward. The RL agent will use a combination of intrinsic and extrinsic reward to learn the QAit objective.

5.2.2 Actor-Critic. We will be using the A2C [24] algorithm trained on a variety of different procedurally generated environments (see section 5.1.1) to promote learning a general policy that can be seen as an actual form of text comprehension compared to simply memorizing states. [28] has shown promising results in A2C's ability to generalise given that the agent is trained on a variety of different but similar environments.

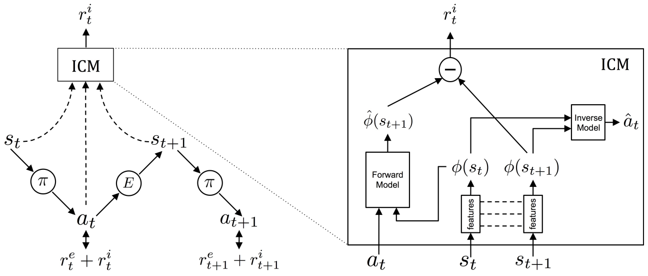

Figure 1. Intrinsic Curiosity Module (ICM) [29]

5.2.3 Testing. The following set of experiments will be trained on the different experiment settings (see table 2) and evaluated using the QAit benchmark testing set:

- Deep Q Networks with an Intrinsic Curiosity Module
- A2C actor-critic agent.
- A2C actor-critic agent with an Intrinsic Curiosity Module.

Ultimately the results found shall be compared against the baselines set out by the QAit task. These include DQN, DDQN, and Rainbow agents as well as purely random agents.

Setting	**Option**
Map Type	Fixed
	Random
RL Agent	DQN
	DDQN
	Rainbow
Number of Games	1
	2
	10
	100
	500
	Unlimited

Table 2. Table of different experiment settings and their values.

5.3 Graph-Based Approach

The focus of this approach is to capture world context in a KG that can be used to overcome the problem of partially observable environments and *catastrophic forgetting*. Which occurs when the model forgets relevant information when presented with new information, which negatively impacts performance [21].

5.3.1 Architecture. We propose integrating a KG that encodes facts and information about the world context relating to the current state of the agent in order to better aid decision making. Specifically, we wish to modify the current architecture of QAit [39] baselines (see Figure 3) by encoding a KG and providing this as additional input to the aggregator.

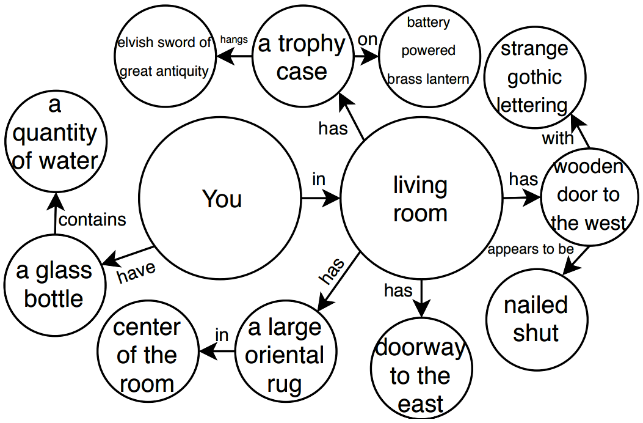

Figure 2. Example of constructed KG [1]

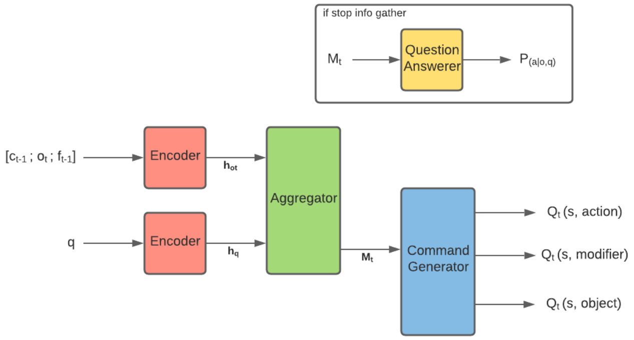

Figure 3. QAit Architecture [39]

The KG will contain information about the world such as the agent's current location and other locations discovered, entities found as well as their discovered attributes (See figure 2). We will be using a graph attention network (GAT) [**?**] to encode the KG in an embedding that can be used by the agent.

5.3.2 Relation Extraction. As text-based games are deemed open-domain QA problem, we will be required to use a relation extractor designed with generalised ontological reasoning. Hence, similarly to work done by Ammanabrolu & Riedl [1], graph information will be extracted using OpenIE [4] in conjunction with some set of predefined heuristics, as outlined by [4].

5.3.3 Testing. The performance of the graph-based methods will be tested by comparing results from the QAit benchmark baselines and the results from the following experiments on the QAit benchmark testing set:

- GB-RL 1: Input GAT embedding directly to aggregator. Run with each setting options in Table 2.
- GB-RL 2: Use KG in the form of a list of triples embedded by observation encoder.

5.4 Sequence Modelling Approach

The process of creating a sequence-to-sequence transformer model to generate action trajectories can be split up into the following parts:

5.4.1 Offline Data. By converting the RL problem into a supervised learning problem, training data is needed ahead of time. This will be done by either heuristic based play-throughs of the environments or simply random roll-outs where each state, action, and reward triple is saved. After every episode play-through, the entire trajectory is processed and stored such as $(R_1, s_1, a_1, R_2, s_2, a_2...R_T, s_T, a_T)$ where R refers to the total summed reward from that timestep onwards, s is a state, and a is an action. The trajectory representation enables the ability for simple auto regressive training. The autoregressive transformer model can be conditioned on reward and starting state to generate the desired action sequence.

5.4.2 Architecture. The transformer is given a context of the last K environmental timesteps which is comprised of the summed reward, state and action tokens. Each token type has it's own linear embedding layer to produce relevant typed embeddings. The tokens are then fed into the transformer model to predict the future action tokens. Each timestep is also embedded into a positional embedding that is added to each token embedding before being used in prediction.

5.4.3 Learning. Using the datasets created from the state, action, and reward trajectories, we will sample batches of sequences having length K, which is a tunable hyper-parameter, and use these batches to train the transformer model with cross-entropy loss. Each prediction head in the transformer is trained to predict the action from the state inputted.

5.4.4 Testing. In order to test the performance of using a sequence modelling approach, we will be using the QAit [39] testing set for the following experiments:

- Comparing small and large context (K) windows when generating actions.
- Using pretrained language model such as GPT or BERT and finetuning on the action generation task.

6 Ethical, Professional and Legal issues

The QAit codebase and baselines provided by Yuan et al. [39] have been made available using an MIT License, allowing for free and unrestricted use. Since QAit is thus open source there is no foreseeable legal issues.

Due to the lack of human or animal subjects in conjunction with having no privacy breaching experiments or data collection, we see that there are no associated ethical concerns.

In terms of professional issues, project members will follow to the Open Source Software guidelines described by [16] and ensure that the use of QAit will be done in such an appropriate professional and ethical manner.

7 Anticipated Outcomes

7.1 Research

7.1.1 Instilling Curiosity and Stochasticity into Text-Based Agents. The addition of curiosity as a form of intrinsic motivation paired with the actor-critic framework is expected to increase the agent's learning speed, training performance and ultimately generalisability on unseen environments.

7.1.2 Graph-Based Approach. Having context embedded into a KG with which the agent utilises will allow for better decision making and improve accuracy. Furthermore, since different domains will contain their own context embedded in their graphs, it is expected to allow the RL agent to produce improved performance in unseen environments.

7.1.3 Sequence Modelling Approach. The anticipated outcome would be that a sequence-to-sequence model will provide equal if not better results than the QAit baseline methods whilst having greater sample efficiency and less training time.

7.2 Impact

7.2.1 Instilling Curiosity and Stochasticity into Text-Based Agents. With the lack of literature present in the IQA task, this research will provide more literature on the use of reinforcement learning as a viable solution to this task as well as contribute to the efficacy of these techniques to create more generalisable agents. Additionally, this research will further illustrate the applicability of these methods to text domains.

7.2.2 Graph-Based Approach. This research will show the potential of using KGs to represent complex and abstracted environments as well as the use of these KGs as supplementary information in an agent's reasoning process.

7.2.3 Sequence Modelling Approach. This research will further show the potential of using sequence-to-sequence models as an abstraction of reinforcement learning in new domains as well as increase the literature on this approaches ability for environment generalisation.

7.3 Key Success Factors

The following are the key success factors pertaining to all three approaches individually as well as their combined effect:

- Successful QAit baseline implementation, use of testing set and environments.
- Extending QAit framework for each individual project task, successful compilation and independent execution.
- Faster and more sample efficient learning.
- Increased zero-shot performance on QAit held data.
- Be able to draw conclusions about RL text-based QA by generating meaningful results.

8 Project Plan

8.1 Risks

A risk assessment matrix is attached in the Appendix (see section A.1).

8.2 Timeline & Milestones

A Gantt chart, containing the timeline for the project along with the project deliverables and milestones, can be found in the Appendix section A.2.

8.3 Deliverables

This project's main deliverable is the final research paper with supporting code for each of the research directions described in this proposal. Other deliverables are:

Deliverable	Due Date
Proposal	24 June
Proposal Presentation	9 July
Software Feasibility Demonstration	10-13 August
Final Paper Draft	6 September
Final Paper	17 September
Final Code Submission	20 September
Final Demonstration	4-8 October
Project Poster	11 October
Project Web Page	18 October

8.4 Resources Required

The primary resources required are as follows:

- UCT High Performance Cluster
- QAit Baselines and Test data
- Access to personal computers and IDEs

Other resources include access to open source software libraries:

- Pytorch
- WandB
- TextWorld

Our work will also seek to extend existing codebases that include, but are not exclusive to, the following:

- Qait Task[1]
- Graph-Based Approach: KG-DQN[2]
- Sequence-Modeling Approach: Decision-Transformer[3]

8.5 Work Allocation

All members of the group will work together on shared deliverables and milestones, such as the presentation and website creation, as well as on the foundational code base before each individual works on their approaches. Edan Toledo will construct an actor-critic agent with curiosity, Roy Cohen will implement a graph-based approach, and Greg Furman will be implementing a sequence model trained with offline learning.

[1] Interactive Language Learning by Question Answering Github (Accessed 4 August 2021)

[2] KG-DQN Github (Accessed 4 August 2021)

[3] Decision-Transformer Github (Accessed 4 August 2021)

References

[1] Prithviraj Ammanabrolu and Matthew Hausknecht. 2020. Graph constrained reinforcement learning for natural language action spaces. *arXiv preprint arXiv:2001.08837* (2020).

[2] Prithviraj Ammanabrolu and Mark O. Riedl. 2019. Playing Text-Adventure Games with Graph-Based Deep Reinforcement Learning. arXiv:1812.01628 [cs.CL]

[3] Prithviraj Ammanabrolu and Mark O Riedl. 2021. Learning Knowledge Graph-based World Models of Textual Environments. *arXiv preprint arXiv:2106.09608* (2021).

[4] Gabor Angeli, Melvin Jose Johnson Premkumar, and Christopher D Manning. 2015. Leveraging linguistic structure for open domain information extraction. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. 344–354.

[5] Marc Bellemare, Sriram Srinivasan, Georg Ostrovski, Tom Schaul, David Saxton, and Remi Munos. 2016. Unifying Count-Based Exploration and Intrinsic Motivation. In *Advances in Neural Information Processing Systems*, D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett (Eds.), Vol. 29. Curran Associates, Inc. https://proceedings.neurips.cc/paper/2016/file/afda332245e2af431fb7b672a68b659d-Paper.pdf

[6] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language Models are Few-Shot Learners. In *Advances in Neural Information Processing Systems*, H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin (Eds.), Vol. 33. Curran Associates, Inc., 1877–1901. https://proceedings.neurips.cc/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf

[7] Yuri Burda, Harrison Edwards, Amos J. Storkey, and Oleg Klimov. 2018. Exploration by Random Network Distillation. *CoRR* abs/1810.12894 (2018). arXiv:1810.12894 http://arxiv.org/abs/1810.12894

[8] Lili Chen, Kevin Lu, Aravind Rajeswaran, Kimin Lee, Aditya Grover, Michael Laskin, Pieter Abbeel, Aravind Srinivas, and Igor Mordatch. 2021. Decision Transformer: Reinforcement Learning via Sequence Modeling. *arXiv preprint arXiv:2106.01345* (2021).

[9] Nuttapong Chentanez, Andrew Barto, and Satinder Singh. 2005. Intrinsically Motivated Reinforcement Learning. In *Advances in Neural Information Processing Systems*, L. Saul, Y. Weiss, and L. Bottou (Eds.), Vol. 17. MIT Press. https://proceedings.neurips.cc/paper/2004/file/4be5a36cbaca8ab9d2066debfe4e65c1-Paper.pdf

[10] Karl Cobbe, Oleg Klimov, Chris Hesse, Taehoon Kim, and John Schulman. 2019. Quantifying Generalization in Reinforcement Learning. In *Proceedings of the 36th International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 97)*, Kamalika Chaudhuri and Ruslan Salakhutdinov (Eds.). PMLR, 1282–1289. http://proceedings.mlr.press/v97/cobbe19a.html

[11] Marc-Alexandre Côté, Ákos Kádár, Xingdi Yuan, Ben Kybartas, Tavian Barnes, Emery Fine, James Moore, Matthew Hausknecht, Layla El Asri, Mahmoud Adada, et al. 2018. Textworld: A learning environment for text-based games. In *Workshop on Computer Games*. Springer, 41–75.

[12] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 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, Volume 1 (Long and Short Papers)*. Association for Computational Linguistics, Minneapolis, Minnesota, 4171–4186. https://doi.org/10.18653/v1/N19-1423

[13] Bhuwan Dhingra, Lihong Li, Xiujun Li, Jianfeng Gao, Yun-Nung Chen, Faisal Ahmed, and Li Deng. 2016. Towards end-to-end reinforcement learning of dialogue agents for information access. *arXiv preprint arXiv:1609.00777* (2016).

[14] Jesse Farebrother, Marlos C. Machado, and Michael Bowling. 2018. Generalization and Regularization in DQN. *CoRR* abs/1810.00123 (2018). arXiv:1810.00123 http://arxiv.org/abs/1810.00123

[15] Daniel Gordon, Aniruddha Kembhavi, Mohammad Rastegari, Joseph Redmon, Dieter Fox, and Ali Farhadi. 2018. Iqa: Visual question answering in interactive environments. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 4089–4098.

[16] F. Grodzinsky, K. Miller, and M. J. Wolf. 2003. Ethical issues in open source software. *J. Inf. Commun. Ethics Soc.* 1 (2003), 193–205.

[17] Matthew Hausknecht, Ricky Loynd, Greg Yang, Adith Swaminathan, and Jason D. Williams. 2019. NAIL: A General Interactive Fiction Agent. arXiv:1902.04259 [cs.AI]

[18] Matthew J. Hausknecht and Peter Stone. 2015. Deep Recurrent Q-Learning for Partially Observable MDPs. *CoRR* abs/1507.06527 (2015). arXiv:1507.06527 http://arxiv.org/abs/1507.06527

[19] Ji He, Jianshu Chen, Xiaodong He, Jianfeng Gao, Lihong Li, Li Deng, and Mari Ostendorf. 2015. Deep reinforcement learning with a natural language action space. *arXiv preprint arXiv:1511.04636* (2015).

[20] Dan Jurafsky and James H Martin. 2014. Speech and language processing. Vol. 3. *US: Prentice Hall* (2014).

[21] Prakhar Kaushik, Alex Gain, Adam Kortylewski, and Alan Yuille. 2021. Understanding Catastrophic Forgetting and Remembering in Continual Learning with Optimal Relevance Mapping. arXiv:2102.11343 [cs.LG]

[22] Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. 2019. Natural Questions: A Benchmark for Question Answering Research. *Transactions of the Association for Computational Linguistics* 7 (March 2019), 452–466. https://doi.org/10.1162/tacl_a_00276

[23] Andrea Madotto, Mahdi Namazifar, Joost Huizinga, Piero Molino, Adrien Ecoffet, Huaixiu Zheng, Alexandros Papangelis, Dian Yu, Chandra Khatri, and Gökhan Tür. 2020. Exploration Based Language Learning for Text-Based Games. *CoRR* abs/2001.08868 (2020). arXiv:2001.08868 https://arxiv.org/abs/2001.08868

[24] Volodymyr Mnih, Adria Puigdomenech Badia, Mehdi Mirza, Alex Graves, Timothy Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. 2016. Asynchronous Methods for Deep Reinforcement Learning. In *Proceedings of The 33rd International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 48)*, Maria Florina Balcan and Kilian Q. Weinberger (Eds.). PMLR, New York, New York, USA, 1928–1937. http://proceedings.mlr.press/v48/mniha16.html

[25] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin A. Riedmiller. 2013. Playing Atari with Deep Reinforcement Learning. *CoRR* abs/1312.5602 (2013). arXiv:1312.5602 http://arxiv.org/abs/1312.5602

[26] Karthik Narasimhan, Tejas Kulkarni, and Regina Barzilay. 2015. Language understanding for text-based games using deep reinforcement learning. *arXiv preprint arXiv:1506.08941* (2015).

[27] OpenAI, Christopher Berner, Greg Brockman, Brooke Chan, Vicki Cheung, Przemysław Dębiak, Christy Dennison, David Farhi, Quirin Fischer, Shariq Hashme, Chris Hesse, Rafal Józefowicz, Scott Gray, Catherine Olsson, Jakub Pachocki, Michael Petrov, Henrique Pondé de Oliveira Pinto, Jonathan Raiman, Tim Salimans, Jeremy Schlatter, Jonas Schneider, Szymon Sidor, Ilya Sutskever, Jie Tang, Filip Wolski, and Susan Zhang. 2019. Dota 2 with Large Scale Deep Reinforcement

Learning. (2019). arXiv:1912.06680 https://arxiv.org/abs/1912.06680

[28] Charles Packer, Katelyn Gao, Jernej Kos, Philipp Krähenbühl, Vladlen Koltun, and Dawn Song. 2018. Assessing Generalization in Deep Reinforcement Learning. *CoRR* abs/1810.12282 (2018). arXiv:1810.12282 http://arxiv.org/abs/1810.12282

[29] Deepak Pathak, Pulkit Agrawal, Alexei A. Efros, and Trevor Darrell. 2017. Curiosity-driven Exploration by Self-supervised Prediction. *CoRR* abs/1705.05363 (2017). arXiv:1705.05363 http://arxiv.org/abs/1705.05363

[30] Tatiana-Andreea Petrache, Traian Rebedea, and Ştefan Trăuşan-Matu. [n.d.]. Interactive language learning-How to explore complex environments using natural language? ([n. d.]).

[31] Jürgen Schmidhuber. 1991. A Possibility for Implementing Curiosity and Boredom in Model-Building Neural Controllers.

[32] Jürgen Schmidhuber. 2010. Formal Theory of Creativity, Fun, and Intrinsic Motivation (1990–2010). *IEEE Transactions on Autonomous Mental Development* 2, 3 (2010), 230–247. https://doi.org/10.1109/TAMD.2010.2056368

[33] Ozgur Simsek, Simon Algorta, and Amit Kothiyal. 2016. Why Most Decisions Are Easy in Tetris—And Perhaps in Other Sequential Decision Problems, As Well. In *Proceedings of The 33rd International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 48)*, Maria Florina Balcan and Kilian Q. Weinberger (Eds.). PMLR, New York, New York, USA, 1757–1765. http://proceedings.mlr.press/v48/simsek16.html

[34] Saku Sugawara, Kentaro Inui, Satoshi Sekine, and Akiko Aizawa. 2018. What makes reading comprehension questions easier? *arXiv preprint arXiv:1808.09384* (2018).

[35] Richard S Sutton and Andrew G Barto. 2018. *Reinforcement learning: An introduction*. MIT press.

[36] Hado van Hasselt, Arthur Guez, and David Silver. 2015. Deep Reinforcement Learning with Double Q-learning. *CoRR* abs/1509.06461 (2015). arXiv:1509.06461 http://arxiv.org/abs/1509.06461

[37] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is All you Need. In *Advances in Neural Information Processing Systems*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (Eds.), Vol. 30. Curran Associates, Inc. https://proceedings.neurips.cc/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf

[38] Xingdi Yuan, Marc-Alexandre Côté, Jie Fu, Zhouhan Lin, Christopher Pal, Yoshua Bengio, and Adam Trischler. 2019. Interactive language learning by question answering. *arXiv preprint arXiv:1908.10909* (2019).

[39] Xingdi Yuan, Marc-Alexandre Côté, Jie Fu, Zhouhan Lin, Chris Pal, Yoshua Bengio, and Adam Trischler. 2019. Interactive Language Learning by Question Answering. 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)*. Association for Computational Linguistics, Hong Kong, China, 2796–2813. https://doi.org/10.18653/v1/D19-1280

[40] Xingdi Yuan, Jie Fu, Marc-Alexandre Cote, Yi Tay, Christopher Pal, and Adam Trischler. 2019. Interactive machine comprehension with information seeking agents. *arXiv preprint arXiv:1908.10449* (2019).

A Appendix

A.1 Risk Assessment

Risk Condition	**Consequence**	**Category**	**Probability**	**Impact**	**Mitigation**	**Monitoring**	**Management**
Hardware Limitations	Processing can take long periods of time and will result in time pressure or practical infeasibility.	Operational	Medium	High	Ensure that all data structures and algorithms used are optimized during development. Ensure all models have GPU support	Run the code base occasionally to ensure that processing isn't taking longer than expected.	Ensure there is sufficient facilities to train models, and options for outsourcing processing.
Drop out of a team member	Losing a team member could result in the collaborative part of the project to be delayed.	Unforeseen	Low	Medium	Ensure good communication between members and take time off when needed.	Regular communication between team members and their physical and mental health.	Ensure that all project tasks have maximum slack/float time to accommodate this risk.
Scope Creep	Project goals will not be met	Management	Low	Medium	Ensure all members are on track with initial timeline and only focus on predetermined goals.	Take note of proposal scope which is approved and ensure that not too much functionality is added beyond it.	Dial back any corollary experiments that are not integral to the project objective.
Team member falls ill with COVID-19 during collaborative phase of project	Having a temporarily missing team member during the collaborative phase will result in delays of the project.	Unforeseen	Medium	Medium	Reduce social contact and adhere to social distancing guidelines.	Be aware of potential COVID-19 symptoms.	Distribute short term work between remaining healthy members for duration of recovery.
Lack of expertise with chosen libraries.	This will result in time wastage as each person attempts to familiarise themselves with the necessary libraries.	Development	Medium	Medium	Dedicate time during vacation to learning the library documentation and tutorials	Check ability to complete library tutorials	Avoid over complication by using the more basic library functionality required to achieve desired network

Table 3. Risk Assessment Matrix

A.2 Gantt Chart

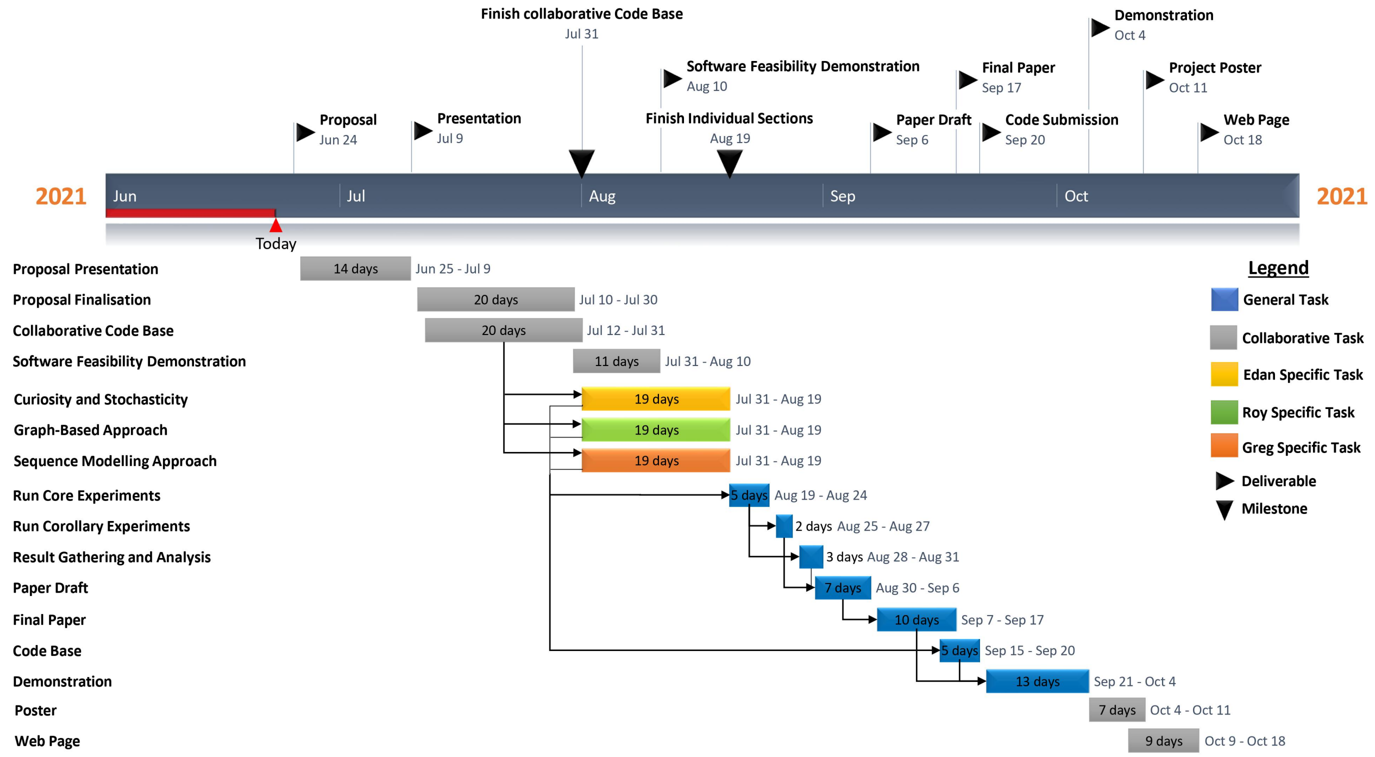

Figure 4. Gantt Chart for IQA project