Graphical modeling for dynamic safety hints generalisation for Safe Deep Reinforcement Learning Agents

Lamogha Chiazor[1] , **Lan Hoang**[1] , **Shivam Ratnakar**[2] and **Minta Siharath**[1]
[1]IBM Research UK
[2]IBM Consulting
{lamogha.chiazor, lan.hoang, minta.siharath}@ibm.com, shirat22@in.ibm.com

Abstract

A major challenge for Deep Reinforcement Learning (DRL) research is to maintain robustness and safety, which is highly important in real-life applications.There is a need to explore AI models/algorithms to utilise system dynamics and their associated reward distribution (assessing risk/opportunities trade-off) to develop algorithms for data-driven Markov Decision Process formulation and safe action sets. Causality modelling is a well-established logical technique which can help identify the effects of interactions of complex evolving systems involving several entities. In this paper, we propose using knowledge graphs which help DRL agents incorporate safety aspects of actions and entities in their decision making process. We dynamically generate safety hints for DRL agents in a text based game environment using a safety concept-net. Our experiments show that DRL agents with safety-hints perform better on safety-based games than agents without them.

1 Introduction

A text-based Reinforcement Learning environment can comprise of multiple components having various dynamic properties such as possible actions and entity states.This is similar to real-life systems like autonomous driving cars, smart homes, smart cities or interactive chat-bots. Entities in these environments may proceed concurrently and interact with each other or external environment.These entities and their properties are subject to modification by other systems. An RL agent trying to explore such environments has to generalise the relationships learned over a set of different scenarios. Generalisation exists in reinforcement learning and machine learning under various forms [Kirk *et al.*, 2021], such as adaptive strategies, logic guidance and succession features [Chen *et al.*, 2023; Hasanbeig *et al.*, 2019; Filos *et al.*, 2021].

However, DRL is an exploratory process by nature, and unsafe actions may be executed especially in the training phase. The safety dynamics may have not been learnt by the agent and the agent may take a large number of action samples in order to learn these constraints.

In this paper, we propose an approach improving safety performance via a type of graphical modelling method for representing the safety causal effects and interactions of DRL agents with other entities in the system they operate. An 'Occurrence network' (ON) in a 'Structured Occurrence network' (SON) is made up of a set of 'condition or state nodes', 'event/action nodes' and 'causality links or edges'. In a SON, multiple ONs are combined using varying types of relationships to represent the dependencies between communicating and evolving sub systems and entities [Li, 2017].

2 Safety Concept Net Graph (SCNG)

Given a DRL agent playing a text-based game, such as the environment TextWorld[1], we propose and hypothesise the use of ON formalism to construct safety concept net graphs which can be used as an additional input combined with commonsense knowledge [Speer *et al.*, 2017] and Natural Language Processing (NLP) techniques to obtain safety hints for the DRL agents to exploit. In this paper, we propose to use causality modelling to capture the dynamics and provide external safety knowledge to the Reinforcement Learning agent to help improve safety performance. We manually construct the SCNG to reflect general household entities (e.g. stove, egg, fridge) that have related safety concerns and occur in several of the text-based games and played by multiple agents.

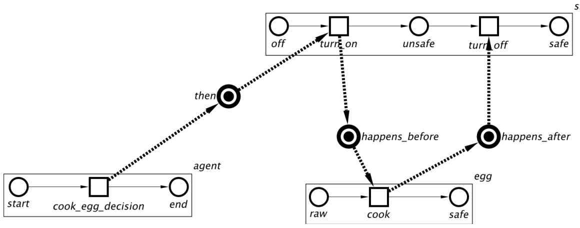

Figure 1: Example of a Safety Concept Net Graph

Figure1 shows an example of a SCNG which can be described as a type of communication SON (CSON). The entities shown in the example are **`stove`**, **`egg`** and **`agent`** and each entity's subgraph is an ON, showing a set of nodes (conditions and actions) and their connecting edges.

[1] https://www.microsoft.com/en-us/research/project/textworld/

An assumption made here, is that in real world settings - entities like a **stove**, irrespective of the type of stove, usually has safety concerns and a set of actions that can be performed on them. The actions might change its state from a safe to an unsafe one and vice versa. Therefore, each ON is made up of at least one or more triplet that represents the Pre Condition → Action → Post Safe/UnSafe Condition that can influence safety in the environment. SCNG is used to represent graphically how each entity synchronously or asynchronously communicates with each other, by connecting the actions from each entity's ON with channel nodes e.g. such as the type of "then", "happens_before" and "happens_after" in figure 1. A synchronous communication channel represents two or more actions that occur at the same time whilst an asynchronous communication channel represents when one of two or more actions must occur first before the others can occur.

3 Generating Safety Hints

Provided with the constructed SCNG and information from the game environment (for example, the current state observations and facts) - we proposed and implemented a safety hints class object, with the methods for obtaining information about:

- the entities of interest in the current state that map to unsafe conditions in the SCNG.
- the safety hints based on the facts from the game environment. An example of a safety hint could be that the **"stove is on and unsafe"** given a fact attribute of "turned_on" for the "stove" in the current state.
- the list of safety hint commands that can address the safety hint concerns. For example, if the safety hint is that the **"stove is on and unsafe"**, then the generated hint command might be **"turn off stove "**. To generate the safety hint commands, we use a combination of *semantic similarity ranking*, *antonyms* and *lemmatisation* to auto construct the hints. For example, if an egg is "raw" in the current state but the stated fact is that the egg is "inedible" - to map the relationship *raw* from our SCNG to the attribute *inedible*, we encode the semantic meaning of both terms "raw" and "inedible", and then compute the cosine similarity between the encoded vectors. Also, the antonym for raw might be "cooked" with a lemma form of "cook". We semantically rank all the possible permitted commands in the current state to the generated hint "cook egg" to help us identify the best safety hint to return.

4 Experiments

The main game we ran has a text-based state and is formulated as a Markov Decision Process for safe exploration. The goal of the DRL agent in the game was to **put a cooked egg in a lunch box**. During the game, the agent gains **100 reward points** for putting the cooked egg into the lunch box and gets penalised by **-10 reward points** for putting a raw egg into the lunch box. The entities in the game were inclusive of a "fridge", a "stove" and an "egg" which can be mapped to entities in a SCNG. This game has clear set of safety actions, hence the reason for it being used as part of the experiments. For each experiment, we ran 500 episodes from start to finish, with three different alterations: without safety hints, with safety hints, and with safety hints and semantic similarity. To obtain different seeds, 3 separate runs per alteration were performed, after which we aggregated the results. This was repeated for three different included modes of the game.

4.1 Results

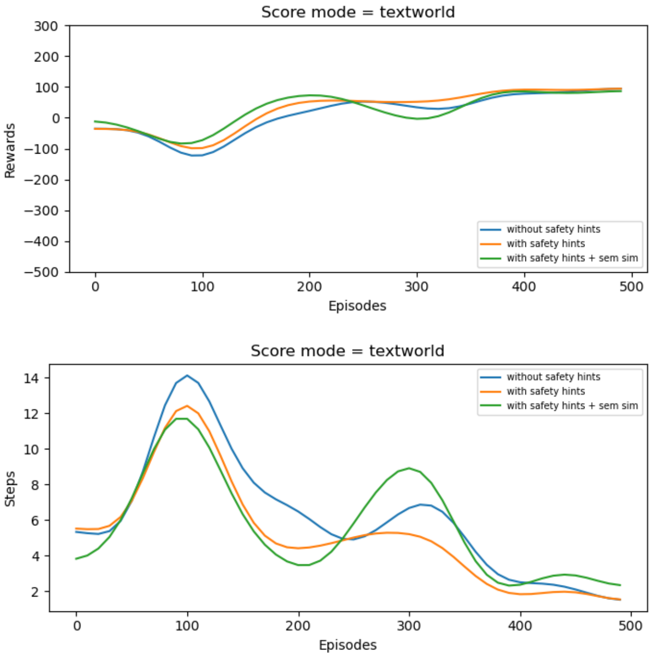

Figure 2: Experiment results showing a knowledge aware agent's performance using safety hints with or without using semantic similarity during training against the baseline with no safety hints.

As shown in 2 the agent with safety hint knowledge tends to accumulate fewer negative rewards compared to the agent without the safety hint knowledge. Since accumulating negative rewards are associated to the unsafe condition, such as "egg" entity been uncooked - the notion here is that, with the safety and semantic hints the agent is automatically exposed to more textual warnings and signs about the environment which in turn helps the agent explore the environment in a safer and more rewarding manner. We also observed with the addition of safety hints that the agent generally uses fewer steps per episode compared to the baseline method of no safety hints, demonstrating a faster learning rate, although adding semantic similarity adds the potential for requiring more steps, demonstrated in the second half of figure 2b.

5 Conclusions

In this paper, we propose to dynamically generate safety hints for DRL agents in a text based game environment to introduce external system dynamic knowledge to a safety-relevant agent which helps to improve the safety performance.

References

[Chen *et al.*, 2023] Yutong Chen, Minghua Hu, Lei Yang, Yan Xu, and Hua Xie. General multi-agent reinforcement learning integrating adaptive manoeuvre strategy for real-time multi-aircraft conflict resolution. *Transportation Research Part C: Emerging Technologies*, 151:104125, 2023.

[Filos *et al.*, 2021] Angelos Filos, Clare Lyle, Yarin Gal, Sergey Levine, Natasha Jaques, and Gregory Farquhar. Psiphi-learning: Reinforcement learning with demonstrations using successor features and inverse temporal difference learning. In *International Conference on Machine Learning*, pages 3305–3317. PMLR, 2021.

[Hasanbeig *et al.*, 2019] Mohammadhosein Hasanbeig, Alessandro Abate, and Daniel Kroening. Certified reinforcement learning with logic guidance. *arXiv preprint arXiv:1902.00778*, 2019.

[Kirk *et al.*, 2021] Robert Kirk, Amy Zhang, Edward Grefenstette, and Tim Rocktäschel. A survey of generalisation in deep reinforcement learning. *arXiv preprint arXiv:2111.09794*, 2021.

[Li, 2017] Bowen Li. *Visualisation and analysis of complex behaviours using structured occurrence nets*. PhD thesis, Newcastle University, 2017.

[Speer *et al.*, 2017] Robyn Speer, Joshua Chin, and Catherine Havasi. Conceptnet 5.5: An open multilingual graph of general knowledge. In *Proceedings of the AAAI conference on artificial intelligence*, volume 31, 2017.