Memory-Based Explainable Reinforcement Learning

Francisco Cruz[1(✉)], Richard Dazeley[1], and Peter Vamplew[2]

[1] School of Information Technology, Deakin University, Geelong, Australia
{francisco.cruz,richard.dazeley}@deakin.edu.au
[2] School of Science, Engineering and Information Technology, Federation University, Ballarat, Australia
p.vamplew@federation.edu.au

Abstract. Reinforcement learning (RL) is a learning approach based on behavioral psychology used by artificial agents to learn autonomously by interacting with their environment. An open issue in RL is the lack of visibility and understanding for end-users in terms of decisions taken by an agent during the learning process. One way to overcome this issue is to endow the agent with the ability to explain in simple terms why a particular action is taken in a particular situation. In this work, we propose a memory-based explainable reinforcement learning (MXRL) approach. Using an episodic memory, the RL agent is able to explain its decisions by using the probability of success and the number of transactions to reach the goal state. We have performed experiments considering two variations of a simulated scenario, namely, an unbounded grid world with aversive regions and a bounded grid world. The obtained results show that the agent, using information extracted from the memory, is able to explain its behavior in an understandable manner for non-expert end-users at any moment during its operation.

Keywords: Reinforcement learning · Explainable reinforcement learning · Human-aligned artificial intelligence

1 Introduction

The aim of reinforcement learning (RL) [17] is to provide an autonomous agent with the ability to learn new skills by only interacting with its environment. RL is a learning approach based on behavioral psychology and conditioned behavior present in mammals and human decision-making within the brain [12]. While RL has been shown to be an effective learning approach, an open issue is the lack of a mechanism that allows them to clearly communicate the reasons why they choose certain actions given a particular state. In this regard, it is not easy for a non-expert end-user to entrust important tasks to an AI-based system that cannot justify its reasoning [1].

J. Liu and J. Bailey (Eds.): AI 2019, LNAI 11919, pp. 66–77, 2019.
https://doi.org/10.1007/978-3-030-35288-2_6

In human cognition, for instance, toddlers are still unable to clearly express reasons about their decisions, mainly due to the incomplete development of language acquisition [14]. The lack of understanding by other interacting agents leads to them not considering toddlers as peers. However, as they develop the ability to give sound and meaningful explanations about their decisions, the mutual confidence level increases and they become collaborative agents[1] [2].

To model artificial systems, different alternatives are possible, i.e., phenomenological models (white-box models), empirical model (black-box models), and hybrid models (gray-box models) [3]. Even though artificial agents are considered to be black-boxes, frequently, it is possible to provide technical clues about why actions are decided, e.g., an RL agent could explain its behavior in terms of Q-values and future reward [4]. Nevertheless, this kind of explanation makes little sense for non-expert users who need to be given explanations using domain-like language in order to allow them to fully understand the agent behavior. In this regard, there have been some research works pursuing a better understanding of RL agent's decisions. However, they have mostly focused on interpretable RL [16] and explainable agency [8], overlooking the option of using the agent's experience to understand its behavior.

In this paper, we propose a memory-based explainable reinforcement learning (MXRL) approach, which allows a learning agent to explain in domain language the decision of selecting an action over the other possible ones. In our approach, explanations are given using the probability of success and the number of transitions needed to reach the goal state. Thus, an RL agent is able to explain its behavior not only in terms of Q-values or the probability of selecting an action but rather in terms of the necessity to complete the intended task.

2 Related Works

2.1 Reinforcement Learning

RL is studied as a decision-making mechanism in both cognitive and artificial agents [17]. An RL agent learns through interaction with its environment, trying to map inputs into actions. In RL, there is no explicit instructor but rather the awareness of how the environment answers to what it is done by the learning agent. Therefore, an agent should be able to sense the environment's state and perform actions in order to transition to a new state.

Formally, an RL agent has to learn a policy $\pi : S \rightarrow A$, where S is the set of states and A the set of available actions, to produce the highest possible reward from a state s_t [17]. The optimal policy is denoted by π^* and the optimal action-value function is denoted by q^*. The optimal action-value function is solved through the Bellman optimality equation for q^*, as shown in Eq. 1.

$$q^*(s_t, a_t) = \sum_{s_{t+1}} p(s_{t+1}|s_t, a_t)[r(s_t, a_t, s_{t+1}) + \gamma \max_{a_{t+1}} q^*(s_{t+1}, a_{t+1})] \quad (1)$$

[1] Agent, in this context, refers to any actor in an environment such as human, animal, or artificial agent.

where s_t is the current state, a_t the taken action, s_{t+1} the next state reached after performing action a_t from the state s_t, and a_{t+1} is an action that could be taken from s_{t+1}. In Eq. 1, p represents the probability of reaching the state s_{t+1} given the current state s_t and the selected action a_t. Finally, r is the reward signal received after performing action a_t from the state s_t.

2.2 Explainable Artificial Intelligence

Over the last few years, explainable artificial intelligence (XAI) has emerged as a prominent research area that aims to provide black-box AI-based systems the ability to give human-like and user-friendly explanations to non-expert end-users [11]. The idea behind XAI is not only intended to provide explanations, but also to allow an AI-system to: justify its decisions and results, control and prevent problems, improve its behavior, and discover new knowledge [1]. The need of XAI is mainly motivated by the need for end-users of trust, interaction, and transparency between them and AI-based systems. Furthermore, XAI is often considered harder than the underlying decision-making process [1], due to the additional interpretability process.

XAI is a vast field, like AI itself, with applications in areas such as transport, finance, medicine, and military among other [6]. Recently, there has been some research studies in explainability pointing to areas such as interpretable RL or explainable agency. These approaches are described next.

2.3 Interpretable Reinforcement Learning

Interpretable RL is an approach which encodes the tasks and actions using human-interpretable instructions. Shu et al. [16] have introduced an approach for hierarchical and interpretable skill acquisition using human descriptions to decompose the tasks into a hierarchical plan with understandable actions. Hein et al. [7] have combined RL with genetic programming (GP) for interpretable policies. They have tested their approach using the mountain car and cart-pole balancing RL benchmarks. However, the provided explanations are only for the learned policy employing equations for that instead of a natural-like representation. Verma et al. [19] have introduced the programmatically interpretable reinforcement learning (PIRL) framework for verifiable agent policies. However, the framework works with symbolic inputs considering only deterministic policies, not including stochastic ones.

In the field of Human-Robot Interaction (HRI), the term of explainable agency has been used to refer to robots engaged in answering questions about its reasons for the decision-making process. Langley et al. [8] propose the elements of explainable agency as content that support explanations, an episodic memory to record states and actions, and access to its experience. However, in their work, they do not implement the proposed approach.

In RL, there have also been a few works trying to provide agents with explanation mechanisms. For instance, Wang et al. [20] proposed an explainable recommendation system using an RL framework. Pocius et al. [13] utilized saliency

maps as a way to explain agent decisions in a partially-observable game scenario. They focused mainly on deep RL and, hence, provided visual explanations. Madumal et al. [10], inspired by cognitive science, proposed to use causal models to derive causal explanations. Nevertheless, the causal model had to be previously known for the specific domain. Sequeira et al. [15] developed a framework to provide explanations employing thoughtful analysis in three levels of the RL agent interaction history. Tabrez and Hayes [18] used an HRI scenario to correct a sub-optimal human model behavior, formulated as a Markov decision process (MDP). In their research, they reported that users found the robot more helpful, useful, and more intelligent when explanations and justification were provided. However, the approach still lacks the comprehensibility of its policy.

3 Memory-Based Explainable Reinforcement Learning

The behavior of an RL agent might be technically explained in terms of the Q-values or also in algorithmic terms. Nonetheless, in this work, we look for explanations that make sense for all kinds of possible end-users and not only to those who are able to understand the underlying learning process behind an artificial agent. In this regard, we look for explanations similarly as it is done by interacting people by using domain-specific language.

To provide artificial agents with the ability to explain the performed actions is currently one of the most critical and complex challenges in future RL research [6]. This challenge is especially important, considering RL-based systems often interact with human observers. Therefore, it is essential that non-expert end-users can understand agents' intentions as well as to obtain more details from the execution in case of a failure [5].

In this paper, we focus on the decision-making process to provide an understanding to the user of what motivates the agent's specific actions from different states, taking into account the problem domain. From a non-expert end-user perspective, we can consider the most relevant questions as to 'why?' and 'why not?' [9,10]. For instance, the following questions may be asked to an artificial agent in order to better understand its behavior:

- Why did you step forward in the last movement?
- Why did you not turn to the right in this situation?

Thus, in order to answer these questions in an understandable domain language, our explanations intend to determine both:

- the artificial agent's probability of success, and
- the number of transitions to reach the goal state, to either finish the task or end it within a time-frame.

Once the probability of reaching the final state is determined the agent will be able to provide the end-user a more compensable explanation for why one action was preferred over others. Moreover, the number of transitions to the goal

will give the end-user an idea about how many steps are necessary to finish the task. Therefore, the agent may explain when an action is preferred to complete the task faster.

We propose a memory-based explainable reinforcement learning (MXRL) approach to compute the success probability P_s and the transitions to the goal N_t consisting of an RL agent with an episodic memory. By accessing the memory, it is possible to understand the agent's behavior based on its experience by using introspection in three levels [15], i.e., environment analysis (to observe certain and uncertain transitions), interaction analysis (to observe state-action frequencies), and meta-analysis (to obtain combined information from episodes and agents). We implement a list of state-action pairs: T_{List} comprising the transactions the agent performed during its learning process.

To compute the success probability P_s, we previously compute the total number of transitions T_t and the number of transitions involved in a success sequence T_s. To obtain T_s, we use the transactions previously saved into the list T_{List}. Every time the agent reaches the final state, we compute the probability $P_s \leftarrow T_s/T_t$ considering transitions involved in the path towards the goal state. The transitions to the goal N_t is computed every time after finishing an episode. For each state, N_t is determined by the position in the list T_{List} since all transitions have been previously saved there. Therefore, each state is as far from the goal as its position in the list, i.e., its index + 1.

Algorithm 1. Memory-based explainable reinforcement learning approach with the on-policy method SARSA to compute the probability of success and the number of transitions to the goal state.

Initialize $Q(s, a)$, T_t, T_s, P_s, N_t
for each episode **do**
    Initialize $T_{List}[]$
    Choose an action using $a_t \leftarrow$ SELECTACTION(s_t)
    repeat
        Take action a_t
        Save state-action transition T_{List}.add(s, a)
        $T_t[s][a] \leftarrow T_t[s][a] + 1$
        Observe reward r_{t+1} and next state s_{t+1}
        Choose next action a_{t+1} using softmax action selection method
        $Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha[r_{t+1} + \gamma Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t)]$
        $s_t \leftarrow s_{t+1}$; $a_t \leftarrow a_{t+1}$
    until s is terminal (goal or aversive state)
    if s is goal state **then**
        for each s,a $\in T_{List}$ **do**
            $T_s[s][a] \leftarrow T_s[s][a] + 1$
        end for
    end if
    Compute $P_s \leftarrow T_s/T_t$
    Compute N_t for each s $\in T_{List}$ as pos(s, T_{List}) + 1
end for

In this paper, an aim is to compare the probability of choosing an action, computed from the Q-values, against the probability of being successful. Therefore, we have implemented the on-policy method SARSA and the softmax action selection method. Algorithm 1 shows our MXRL approach to train RL agents using episodic memory. Whereas in line 7 each executed state-action pair is saved into the memory, lines 19 and 20 compute the final probabilities of success P_s and the number of transitions to the goal state N_t for each episode.

4 Experimental Set-Up

In order to produce explanations related to the context, we implemented a grid world scenario in two versions: bounded and unbounded. Therefore, the same state-action pair may lead to different characteristic for the explanation depending on the context. We use a 3×4 grid world, as shown in Fig. 1. In the figure, it is possible to observe the 12 states in which the agent can be. The goal state is shown with a green circle at the right bottom. The gray circle represents the agent which needs to find one path towards the goal state. In every episode, the agent is located in a random initial position within the grid world. Over the episodes, the learning agent has to learn a policy in order to reach the goal position. There are four allowed actions in this scenario: down, up, right, and left.

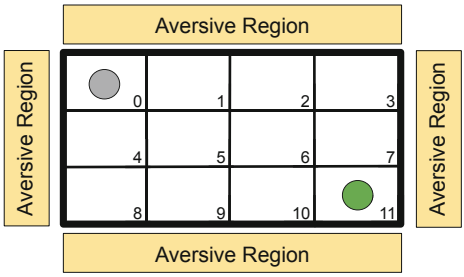


Fig. 1. The 3×4 grid world surrounded by aversive regions. The agent may move in four directions: down, up, right, and left. The green circle shows the goal state. If the aversive region is reached by the agent, the learning episode is finished and a new one started. In the bounded grid world scenario, the agent is not allowed to step into the aversive regions. (Color figure online)

In principle, we consider an unbounded grid world, i.e., a grid world where the agent might get into aversive regions leading to stop the current learning episode and restart a new one. The aversive regions are shown in yellow in Fig. 1. In this case, the probability of being successful is computed after every learning episode and depends on the experience of each agent to reach the final state.

Furthermore, we have also considered a bounded grid world, i.e., a grid world from where the agent is not allowed to step out. Therefore, every time the agent

tries to step out the grid world, the current state is not updated, keeping the position as it was previously to select that action. In this context, the agent has a constant success probability of 1 since it is always able to complete the task. However, the time steps needed to get the goal are different for each reached state after performing an action.

5 Experimental Results

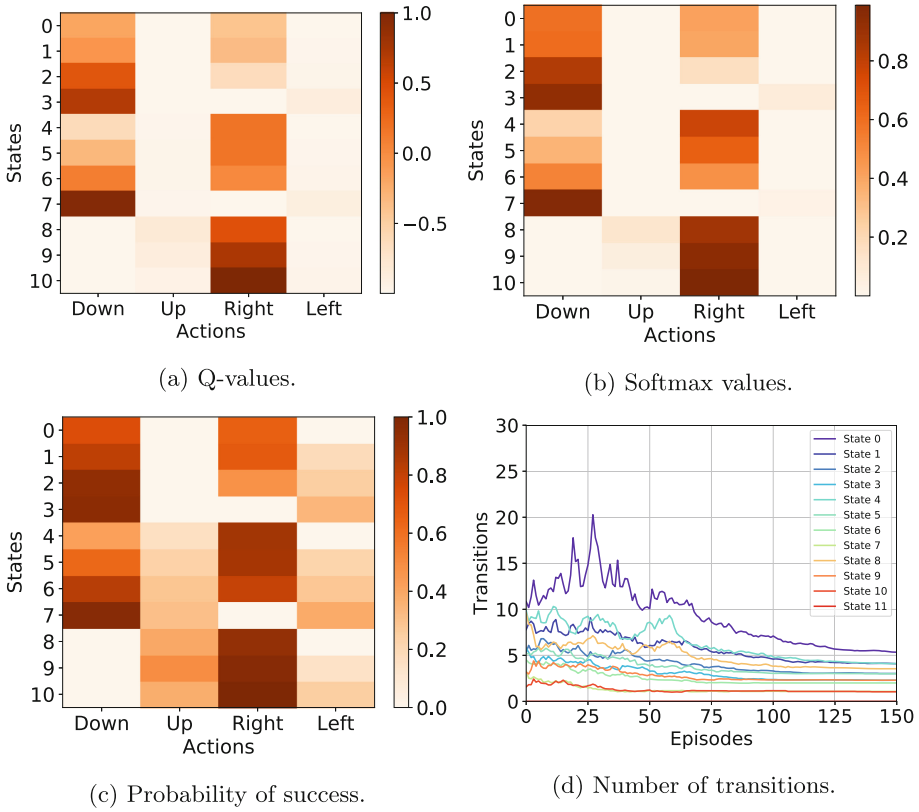
For the learning process, the reward function returns a positive reward of 1 when the agent reaches the final state and a negative reward of -1 when the agent enters an aversive region. All the experiments have been performed using the on-policy learning algorithm SARSA and the softmax action selection method for the training of 100 agents. The following plots show the average results. The parameters used for the training are: learning rate $\alpha = 0.3$, discount factor $\gamma = 0.9$, and softmax temperature $\tau = 0.25$, all of them were experimentally determined and related to our scenario. The previous parameters are mentioned here just as a reference, but they are not relevant for this work. These parameters do affect the agents' ability to learn a solution. However, we are interested in understanding the decision, rather than the speed or capacity of the learning agents.

5.1 Unbounded Grid World

In the unbounded grid world scenario, the agent is allowed to step out of the grid into the aversive region. Figure 2 shows the obtained Q-values, the probability of choosing an action, the probability of success, and the number of transitions to the goal state.

After training is complete, the average Q-values are shown in Fig. 2a. It can be observed that the agent does not favor actions of going up or going left since, independently of the current state, they always result in the agent moving further away from the goal state. In general terms, the Q-values, also show symmetric values, which indeed means the agent may select any route to the goal as long as its movements are down or right. Of course the closer to the goal state the higher reward which is shown, for instance, in states 7 and 10 with actions down and right respectively, both cases being final state's neighbors. There are a few exceptions with low Q-value when moving down (states 8, 9, and 10) and moving right (states 3 and 7) which represent the fact of stepping out the grid into the aversive region.

Figure 2b shows the average softmax probability of choosing an action from each state after learning. Although the probabilities of choosing an action are connected with the Q-values in terms of the different possible paths to the goal state, they only explain how likely it is to select an action rather than how successful the agent will be by selecting it. Thus, it cannot clearly be explained yet to a non-expert end-user why an RL agent would favor one of those actions.



(a) Q-values.

(b) Softmax values.

(c) Probability of success.

(d) Number of transitions.

Fig. 2. Obtained results unbounded grid world. (a) Q-values. It can be seen that the agent does not favor actions which lead it further from the goal state, i.e., moving up or left. Additionally, the Q-values show symmetry considering the possible paths to the goal state. (b) Probability of choosing an action. While the softmax values show that the agent may select any path to the goal state with similar probability, they do not provide enough information in domain language. (c) Probability of success considering state-action pairs. Actions leading to the aversive region have a probability of success equal to 0. Moreover, actions far from the goal state or actions which get the agent further from the goal may also be successful if the right sequence is taken from there. (d) Evolution of the number of transitions over the learning episodes to reach the goal state. After training, the agent learns the shortest path to the goal.

Figure 2c shows the probability of success for each state-action pair after the learning process. The probabilities are computed after each episode using the memory. As previously discussed, they are a more transparent manner to explain to a non-expert end-user the reasons why an RL agent favors specific actions from specific states. In Fig. 2c, for instance, it is clear to see what actions lead to the aversive region as they show probability equal to 0. Moreover, it is shown that even actions which move the agent further from the goal state may

eventually be successful, or that states located far from the goal may also be highly successful if the proper sequence of actions is taken.

Additionally, Fig. 2d shows the number of transitions to reach the goal position from every state over the learning episodes. The number of actions executed in this case is computed taking into account only the successful runs of RL. After 150 episodes, the agent learns the shortest possible paths from all states.

In this context, one possible question to the artificial agent is: Why did you choose action down when in state 0? Trying to explain this in term of Q-values means to show to an end-user the following information. $Q(s = 0, a = \text{down}) = -0.181$, $Q(s = 0, a = \text{up}) = -0.998$, $Q(s = 0, a = \text{right}) = -0.411$, $Q(s = 0, a = \text{left}) = -0.998$, which is pointless for a non-expert user. However, if we use the probability of success, we can observe that $P_s(s = 0, a = \text{down}) = 0.736$, $P_s(s = 0, a = \text{up}) = 0$, $P_s(s = 0, a = \text{right}) = 0.656$, $P_s(s = 0, a = \text{left}) = 0$. Therefore, the agent may answer the end-user: I chose to go down because that has a 73.6% probability of successfully reaching the goal. Another possible question to the agent is: Why did you not choose to go left when in state 0? Given the previous P_s values, one possible answer is: I did not choose left because that has a zero probability of success, whereas by choosing down has a 73.6% probability of success, which was higher than other actions.

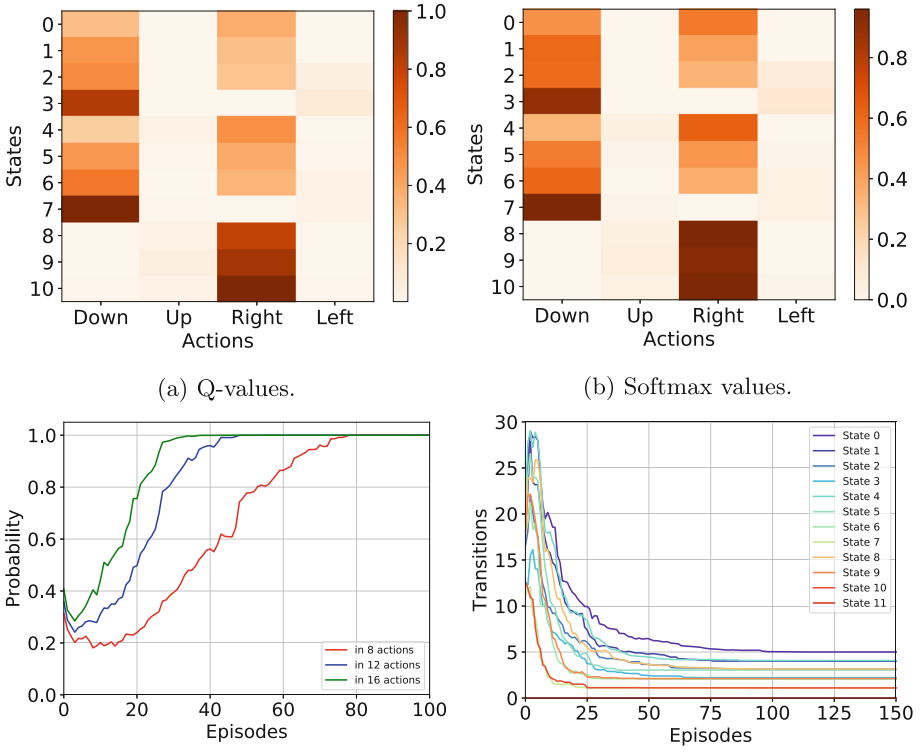
5.2 Bounded Grid World

As aforementioned, the bounded grid world is an always success scenario since the agent cannot step out of the grid world into the aversive region and, therefore, eventually will always reach the goal state. Figure 3 shows the obtained Q-values, probability of choosing an action, and the number of actions to the final state.

In Fig. 3a, the obtained Q-values present similar distribution as the previous unbounded case, i.e., actions moving the agent up and left have lower values in comparison with down and right that moves the agent closer to the goal position.

In this case, the probability of choosing an action is also related to the Q-values, as shown in Fig. 3b. However, this probability does not provide enough information to understand and explain the action-selection decision by the RL agent, especially considering that the agent never fails the task in the bounded grid world. Therefore, in this scenario, to compute the number of transitions to reach the goal and the probability of success within a time window is imperative. Thus, an RL agent may answer more clearly questions as to why a particular action is preferred over others from a specific state referring to the number of steps needed to reach the goal.

Figure 3c shows the evolution of the probability of success over the learning episodes with the agent starting in position 0 (similar charts can be generated starting from any state). Three different time windows are considered as examples, i.e., the probability of reaching the goal in 8, 12, and 16 actions. In Fig. 3d is shown the number of transitions from each state to reach the goal state over the learning episodes. The RL agent may use this information to answer if a taken action reaches another state, from where it is faster to get it to the final state.



(a) Q-values.

(b) Softmax values.

(c) Probability of Success.

(d) Number of transitions.

Fig. 3. Obtained results bounded grid world. (a) Q-values. The RL agent favors down and right actions since get it closer to the goal state. As in the unbounded scenario, the Q-values are symmetric meaning that the agent has no particular preference for similar paths to the goal. (b) Probability of choosing an action. The softmax probabilities show only how likely it is to select an action after the learning process; however, they do not present information about the time-steps needed to success from a particular state-action pair. (c) Probability of success from position 0 within a specific window of actions using cumulative normal distribution. The larger the window, the higher the probability of finishing the task. To obtain the maximal probability are required 78, 48, and 32 episodes for a window of 8, 12, and 16 actions respectively. (d) Evolution of the number of transitions to reach the goal state. Since this is an always success scenario, it is relevant to provide explanations about the steps needed to reach the goal.

In this problem, a possible question for the agent could be: What is the probability of finishing the task in 8 movements starting from the state 0? One more time, if we want to answer this question in terms of Q-vales to the end- user we should show that $Q(s = 0, a = \text{down}) = -0.368$, $Q(s = 0, a = \text{up}) = -0.993$, $Q(s = 0, a = \text{right}) = -0.243$, $Q(s = 0, a = \text{left}) = -0.994$, which has no meaning for a non-expert end-user. However, if we refer to the plot Fig. 3c, we can clearly observe the probability of finishing the task in 8 movements starting

from state 0. For instance, after 30 training episodes the agent may answer: I can finish the task in 8 movements with a probability of 39.4%, or after 60 episodes the agent's answer may be: I can complete the task in 8 moves with a probability of 86.5%.

6 Conclusions

In this work, we have presented an MXRL approach aiming an agent to explain to non-expert end-users the reasons why some decisions are taken in certain situations. To this end, using a episodic memory, we have computed the probability of success and the number of steps to the goal state, which allow the agent to provide explanations using domain-based language. Our experiments have been performed in a scenario with two variations, an unbounded and a bounded grid world. The obtained results show that the agent, using the episodic memory, is able to find clear explanations for end-users with no previous knowledge of machine learning techniques.

The explanations shown in this work are examples of possible answers obtained from the resulting probability of success and the number of transitions to the goal during the learning process. Currently, our method presents some limitations as the use of memory in large solution spaces. Moreover, to this point in this work, we have only considered a discrete episodic task with a terminal goal state. In this regard, the obtained results motivate future work in many possible directions. For instance, we are planning to extend our approach to compute the probability of success and the number of transitions to the goal by using another more general method, such as function approximator, Bayesian methods, or phenomenological relations from the Q-values. By using a more general estimation method, our approach might be scaled to more complex scenarios as problems with no final state, i.e., which need to operate continuously, or problems with continuous state-action representation.

References

1. Adadi, A., Berrada, M.: Peeking inside the black-box: a survey on explainable artificial intelligence (XAI). IEEE Access **6**, 52138–52160 (2018)
2. Conrad, B., Gross, D., Fogg, L., Ruchala, P.: Maternal confidence, knowledge, and quality of mother-toddler interactions: a preliminary study. Infant Mental Health J. **13**(4), 353–362 (1992)
3. Cruz, F., Acuña, G., Cubillos, F., Moreno, V., Bassi, D.: Indirect training of grey-box models: application to a bioprocess. In: Liu, D., Fei, S., Hou, Z., Zhang, H., Sun, C. (eds.) ISNN 2007. LNCS, vol. 4492, pp. 391–397. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-72393-6_47
4. Cruz, F., Magg, S., Nagai, Y., Wermter, S.: Improving interactive reinforcement learning: what makes a good teacher? Connect. Sci. **30**(3), 306–325 (2018)
5. Dulac-Arnold, G., Mankowitz, D., Hester, T.: Challenges of real-world reinforcement learning. arXiv preprint arXiv:1904.12901 (2019)

6. Gunning, D.: Explainable artificial intelligence (XAI). Defense Advanced Research Projects Agency (DARPA), nd Web (2017)
7. Hein, D., Udluft, S., Runkler, T.A.: Interpretable policies for reinforcement learning by genetic programming. Eng. Appl. Artif. Intell. **76**, 158–169 (2018)
8. Langley, P., Meadows, B., Sridharan, M., Choi, D.: Explainable agency for intelligent autonomous systems. In: Twenty-Ninth IAAI Conference, pp. 4762–4763 (2017)
9. Lim, B.Y., Dey, A.K., Avrahami, D.: Why and why not explanations improve the intelligibility of context-aware intelligent systems. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, pp. 2119–2128. ACM (2009)
10. Madumal, P., Miller, T., Sonenberg, L., Vetere, F.: Explainable reinforcement learning through a causal lens. arXiv preprint arXiv:1905.10958 (2019)
11. Miller, T.: Explanation in artificial intelligence: insights from the social sciences. Artif. Intell. **267**, 1–38 (2018)
12. Niv, Y.: Reinforcement learning in the brain. J. Math. Psychol. **53**, 139–154 (2009)
13. Pocius, R., Neal, L., Fern, A.: Strategic tasks for explainable reinforcement learning. In: The Thirty-Third AAAI Conference on Artificial Intelligence (AAAI 2019), p. 2 (2019)
14. Robertson, S.B., Weismer, S.E.: Effects of treatment on linguistic and social skills in toddlers with delayed language development. J. Speech Lang. Hearing Res. **42**(5), 1234–1248 (1999)
15. Sequeira, P., Yeh, E., Gervasio, M.T.: Interestingness elements for explainable reinforcement learning through introspection. In: IUI Workshops, p. 7 (2019)
16. Shu, T., Xiong, C., Socher, R.: Hierarchical and interpretable skill acquisition in multi-task reinforcement learning. arXiv preprint arXiv:1712.07294 (2017)
17. Sutton, R.S., Barto, A.G.: Reinforcement Learning: An Introduction. Bradford Book, Cambridge (1998)
18. Tabrez, A., Hayes, B.: Improving human-robot interaction through explainable reinforcement learning. In: 2019 14th ACM/IEEE International Conference on Human-Robot Interaction (HRI), pp. 751–753. IEEE (2019)
19. Verma, A., Murali, V., Singh, R., Kohli, P., Chaudhuri, S.: Programmatically interpretable reinforcement learning. arXiv preprint arXiv:1804.02477 (2018)
20. Wang, X., Chen, Y., Yang, J., Wu, L., Wu, Z., Xie, X.: A reinforcement learning framework for explainable recommendation. In: 2018 IEEE International Conference on Data Mining (ICDM), pp. 587–596. IEEE (2018)