Efficient Text-based Reinforcement Learning by Jointly Leveraging State and Commonsense Graph Representations

Anonymous ACL-IJCNLP submission

Abstract

Text-based games (TBGs) have emerged as useful benchmarks for evaluating progress at the intersection of grounded language understanding and reinforcement learning (RL). Recent work has proposed the use of external knowledge to improve the efficiency of RL agents for TBGs. In this paper, we posit that to act efficiently in TBGs, an agent must be able to track the state of the game while retrieving and using relevant commonsense knowledge. Thus, we propose an agent for TBGs that induces a graph representation of the game state and jointly grounds it with a graph of commonsense knowledge from ConceptNet. This combination is achieved through *bidirectional knowledge graph attention* between the two symbolic representations. We show that agents that incorporate commonsense into the game state graph outperform baseline agents.

1 Introduction

Text-based games (TBGs) are simulation environments in which an agent interacts with the world purely in the modality of text. TBGs have emerged as key benchmarks for studying how reinforcement learning agents can tackle the challenges of language understanding, partial observability, and action generation in combinatorially-large action spaces. One particular text-based gaming environment, TextWorld (Côté et al., 2018), has received significant attention in recent years.

Recent work has shown the need for additional knowledge to tackle the challenges in TBGs. Ammanabrolu and Riedl (2019) proposed handcrafted rules to represent the current state of the game using a state knowledge graph (much like a map of the game). Murugesan et al. (2021) proposed an extension of TextWorld, called TextWorld Commonsense (TWC), to test agents' ability to use commonsense knowledge while interacting with the world. The hypothesis behind TWC is that

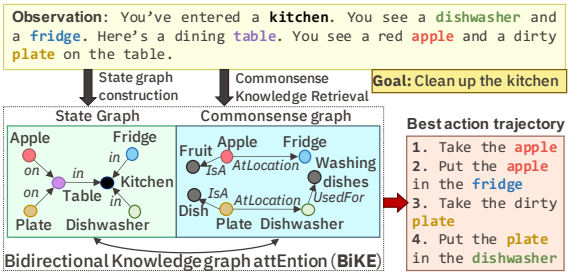


Figure 1: An illustration of a text-based game that requires both the state representation of the game as well as the external commonsense knowledge for efficient exploration and learning the best action trajectory. The observation text feeds into the state and commonsense graphs; and the best action trajectory is computed based on information from both graphs.

commonsense knowledge allows the agent to understand how current actions might affect future world states; and enable look-ahead planning (Juba, 2016), thus leading to sample-efficient selection of actions at each step and driving the agent closer to optimal performance.

In this paper, we posit that to efficiently act in such text-based gaming environments, an agent must be able to effectively track the state of the game, and use that to jointly retrieve and leverage the relevant commonsense knowledge. For example, commonsense knowledge such as `apple should be placed in the refrigerator` would help the agent to act closer to the optimal behavior; whereas state information like `apple is on the table` would help the agent plan more efficiently. Thus, we propose a technique to: (a) track the state of the game in the form of a symbolic graph that represents the agent's current belief of the state of the world (Ammanabrolu and Hausknecht, 2020; Adhikari et al., 2020); (b) retrieve the relevant commonsense knowledge from ConceptNet (Speer et al., 2017), and (c) jointly leverage the state graph and the retrieved commonsense graph. This combined information is then used to select the optimal action. Finally, we demonstrate the performance of our agent against state of the art baseline agents on the TWC Environment.

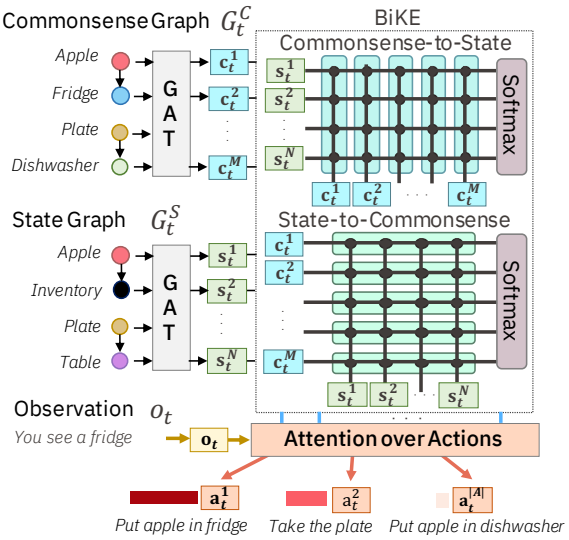


Figure 2: Visualization of our overall approach with BiKE.

2 Model & Architecture

TBGs can be framed as partially observable Markov decision processes (POMDPs) (Spaan, 2012) denoted $\langle S, A, O, T, E, r \rangle$, where: S denotes the set of states, A denotes the action space, O denotes the observation space, T denotes the state transition probabilities, E denotes the conditional observation emission probabilities, and $r : S \times A \rightarrow \mathbb{R}$ is the reward function. The observation o_t at time step t depends on the current state. Both observations and actions are rendered in text. The agent receives a reward at every time step t: $r_t = r(o_t, a_t)$, and the agent's goal is to maximize the expected discounted sum of rewards: $\mathbb{E}[\sum_t \gamma^t r_t]$, where $\gamma \in [0, 1]$ is a discount factor.

The high-level architecture of our model contains three major components: (a) the input encoder; (b) a graph-based knowledge extractor; and (c) the action prediction module. The input encoding layers are used to encode the observation o_t at time step t and the list of admissible actions using GRU (Ammanabrolu and Hausknecht, 2020). The graph-based knowledge extractor extracts relevant knowledge from complementary knowledge sources: the game state, and external commonsense knowledge. We allow information from each knowledge source to guide and direct better representation learning for the other.

Recent efforts have demonstrated the use of primarily two different types of knowledge sources for TextWorld RL Agents. A **State Graph** (SG) captures state information (Ammanabrolu and Riedl, 2019) about the environment represented via a language-based semantic graph. The example in Figure 2 shows that information such as `Apple` → `on` → `Table` is extracted from the textual observations from the environment. Specifically, Ammanabrolu and Riedl (2019) create such knowledge graphs by extracting information using OpenIE (Angeli et al., 2015) and some manual heuristics. A **Commonsense Graph** (CG) captures external commonsense knowledge (Murugesan et al., 2021) between entities (from commonsense knowledge sources such as ConceptNet). For instance, the CG can encode information such as `Apple` → `LocatedIn` → `Fridge` from ConceptNet; whereas the SG cannot capture such information since it is restricted to the textual observations from the environment. We posit that RL agents can make use of information from both these graphs during different sub-tasks, enabling efficient learning. The SG provides the agent with a symbolic way of representing its current perception of the game state, including its understanding of the surroundings. On the other hand, the CG provides the agent with complementary human-like knowledge about what actions make sense in a given state, thus enabling more efficient exploration of the very large natural language based action space.

We combine the state information with commonsense knowledge using a **Bi**directional **K**nowledge-graph att**E**ntion (**BiKE**) mechanism, which re-contextualizes the *state* and *commonsense* graphs based on each other for optimal action trajectories. Figure 2 provides a compact visualization.

3 Knowledge Integration using BiKE

The aforementioned graph-based knowledge extractor produces M entities $(c_t^1, c_t^2, \cdots, c_t^M)$ for the commonsense graph (CG); and N entities $(s_t^1, s_t^2, \cdots, s_t^N)$ for the state graph (SG). Note that the entities extracted for the CG are based on the vocabulary used in ConceptNet, and may not necessarily have the same set of entities as the SG (Figure 1). We embed the extracted entities in both graphs using *Numberbatch* (Liu and Singh, 2004). We then encode these graph representations using a Graph Attention Network (GAT) (Veličković et al., 2018). GAT allows the node entities s_t and c_t within the graphs G_t^S and G_t^C respectively to share information among each other by message passing.

We then integrate sub-graphs extracted from the previous steps to improve the agent's exploration strategy. Inspired from bidirectional attention mechanism in QA (Seo et al., 2016), we use BiKE attention mechanism between G_t^S and G_t^C to

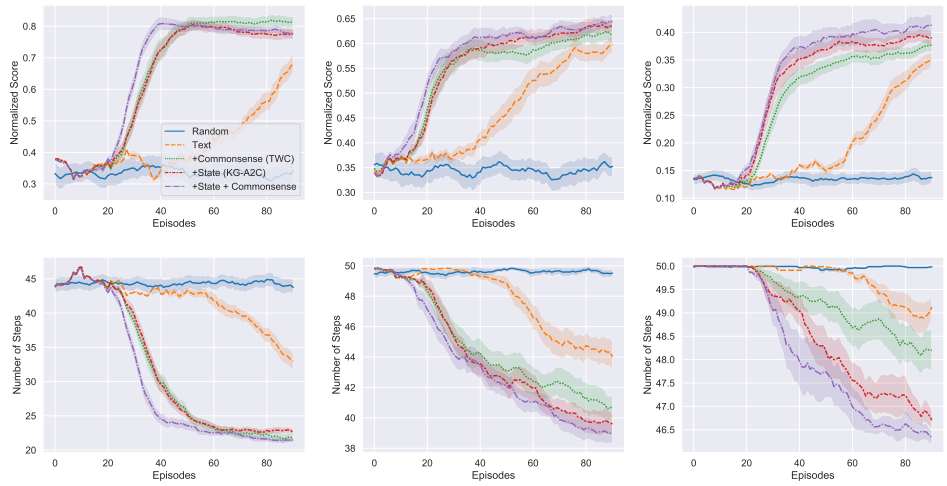


Figure 3: Performance evaluation (showing mean and standard deviation averaged over 3 runs) for the three difficulty levels: Easy (left), Medium (middle), Hard (right) using normalized score and the number of steps taken.

fuse the knowledge from these two graphs. The information flow across the graphs allows the model to learn commonsense-aware state graph representations, and state-aware commonsense knowledge graph representations.

To implement this, we compute a graph similarity matrix $\mathtt{S} \in \mathbb{R}^{N \times M}$ across the graph entities to learn a state-to-commonsense graph attention function and a commonsense-to-state graph attention function. $\mathtt{S}_{ij} = f(s_t^i, c_t^j)$ captures how each node s_t^i in the graph G_t^S is linked to a node c_t^j in the other graph G_t^C, and vice versa. Here f is a learnable function that maps s_t^i and c_t^j to a similarity score. This allows us to measure the similarity between (for instance) `Apple` observed in the state graph and `Apple` observed in the commonsense graph. We compute the state-to-commonsense graph attention values A by taking a softmax along the rows of $\mathtt{S}$: this signifies the attention bestowed by each state graph node on the nodes of the commonsense graph. Similarly, we compute the commonsense-to-state graph attention values $\bar{A}$ by taking a softmax along the columns of $\mathtt{S}$. We capture the relevant knowledge in the commonsense graph G_C^t by updating the state representations $\bar{s}_t^i$. We compute the updated state representation as: $s_{t+1}^i = g(s_t^i, \tilde{s}_t^i, \bar{s}_t^i)$; where $\tilde{s}_t^i = \sum_j A^{ij} c_t^j$, $\bar{s}_t^i = \sum_j A^{ij} \sum_{i'} \bar{A}^{ji'} s_t^{i'}$, and g is a learnable function that maps the concatenated s_t^i, $\tilde{s}_t^i$, and $\bar{s}_t^i$ to an updated state representation. Finally, we use the general attention between the o_t and the state graph entities s_{t+1} to get the state graph representation $\mathbf{g}_{t+1}^S$ (Luong et al., 2015). We perform a similar process for the commonsense-to-state graph attention and obtain the commonsense graph representation: $\mathbf{g}_{t+1}^C$. We select the relevant action by computing an attention over the actions: $h(o_t, a_t^i, \mathbf{g}_{t+1}^S, \mathbf{g}_{t+1}^C)$; where h is a learnable function that projects the concatenation $\langle\, o_t, a_t^i, \mathbf{g}_{t+1}^S, \mathbf{g}_{t+1}^C \,\rangle$ to the attention score for the i^{th} action.

4 Experiments

We generate a set of games with 3 difficulty levels using the TWC (Murugesan et al., 2021) framework[1]: (i) *easy* level, which has 1 room containing 1 to 3 objects; (ii) *medium* level, which has 1 or 2 rooms with 4 or 5 objects; and (iii) *hard* level, a mix of games with a high number of objects (6 or 7 objects in 1 or 2 rooms) or high number of rooms (3 or 4 rooms containing 4 or 5 objects).

We compare 5 text-based RL agents: (a) a text-only agent (**Text**), which selects the best action based only on the encoding of the history of observations; (b) **DRRN** (He et al., 2016; Narasimhan et al., 2015), which relies on the relevance between the observation and action spaces; (c) an agent enhanced with access to an external commonsense knowledge graph (**+Commonsense**) (Murugesan et al., 2021); (d) an agent that, following (Ammanabrolu and Hausknecht, 2020), models the state of the world as a symbolic graph (**+State**); and (e) the agent (BiKE) described in Section 2, which relies on both state and commonsense graph representations. The agents are trained over 100 episodes with a 50-step maximum. All policies are learned using Actor-Critic (Mnih et al., 2016; Adolphs and Hofmann, 2019).

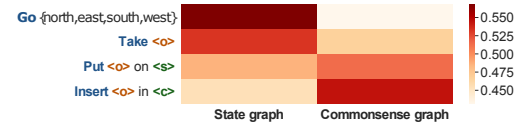
4.1 Improving Performance with State and Commonsense Knowledge

Figure 3 shows the learning curves for the text-only agent and the agents equipped with state and/or

[1] https://github.com/IBM/commonsense-rl

		Easy		Medium		Hard	
		#Steps	Norm. Score	#Steps	Norm. Score	#Steps	Norm. Score
IN	Text	23.83 ± 2.16	0.88 ± 0.04	45.90 ± 0.22	0.60 ± 0.02	49.84 ± 0.38	0.30 ± 0.02
	DRRN	22.08 ± 4.17	0.82 ± 0.06	45.18 ± 1.19	0.59 ± 0.02	49.82 ± 0.61	0.29 ± 0.01
	+Commonsense (TWC)	20.59 ± 5.01	0.89 ± 0.06	44.89 ± 1.52	0.62 ± 0.03	48.45 ± 1.13	0.32 ± 0.04
	+State (KG-A2C)	22.10 ± 2.91	0.86 ± 0.06	43.05 ± 2.52	0.62 ± 0.03	48.00 ± 0.61	0.32 ± 0.00
	+State + Commonsense (BiKE)	18.27 ± 1.13	0.94 ± 0.02	41.01 ± 1.61	0.64 ± 0.02	47.19 ± 0.64	0.34 ± 0.02
OUT	Text	29.90 ± 2.92	0.78 ± 0.02	44.08 ± 0.93	0.55 ± 0.01	50.00 ± 0.00	0.20 ± 0.02
	DRRN	29.71 ± 1.81	0.76 ± 0.05	44.04 ± 1.64	0.56 ± 0.02	50.00 ± 0.00	0.21 ± 0.02
	+Commonsense (TWC)	27.74 ± 4.46	0.78 ± 0.07	42.61 ± 0.65	0.58 ± 0.01	50.00 ± 0.00	0.19 ± 0.03
	+State (KG-A2C)	28.34 ± 3.63	0.80 ± 0.07	41.61 ± 0.37	0.59 ± 0.01	50.00 ± 0.00	0.21 ± 0.00
	+State + Commonsense (BiKE)	25.59 ± 1.92	0.83 ± 0.01	39.34 ± 0.72	0.61 ± 0.01	50.00 ± 0.00	0.23 ± 0.02

Table 1: Test-set performance results for within distribution (*IN*) and out-of-distribution (*OUT*) games.



(a) Average relevance of the main action templates to the state and commonsense graphs across the *hard* games.

Timestep	t	$\Longrightarrow$ $t+1$	$\Longrightarrow$ $t+2$
Room	*Living Room*	$\Longrightarrow$ *Living Room*	$\Longrightarrow$ *Bedroom*
Action Taken	take checkered jumper	$\Longrightarrow$ go west	$\Longrightarrow$ insert checkered jumper into wardrobe
Most relevant graph	*State graph*	*State graph*	*Commonsense graph*
Most relevant nodes	checkered jumper	checkered jumper, exit to west	wardrobe

(b) Example of most relevant graphs and nodes (by action taken) for one example game excerpted from the *hard* difficulty level.

Figure 4: Analysis of the relevance given to the: (a) state and commonsense graphs; and to (b) their nodes (by action taken).

commonsense graph representations at training time. For reference, we also report the performance of an agent that selects a random action at each time step (**Random**). We notice that, overall, agents equipped with either state or commonsense graph representations perform better than their text-only counterparts, both in terms of the number of steps taken and the normalized score. In particular, the BiKE agent outperforms all other agents in all difficulty levels, showing that symbolic state representations and prior commonsense knowledge can be jointly used for better sample efficiency and results. Table 1 shows the performance of the agents on the test set. Following Murugesan et al. (2021), we compared our agents on two test sets: (**IN**) uses the same entities as the training set, and (**OUT**) uses entities that were not included in the training set. The experimental results show that the BiKE agent generalizes better than all the baselines across the 3 difficulty levels.

4.2 Qualitative Analysis

From Figure 3 and Table 1, we notice that the **+Commonsense** agent performs better on the *easy* level, whereas the **+State** agent performs better on the *medium* and *hard* levels. This suggests that the state representation can be leveraged to drive exploration and interaction with objects in environments with multiple rooms; whereas prior commonsense knowledge allows the agent to act more efficiently by selecting the appropriate commonsensical locations of different objects. In order to investigate this hypothesis, we computed the average importance given by the agent to the state graph and the commonsense graph when selecting the different action templates shown in Figure 4a. For each action template, the figure shows the normalized attention weight given to the two graphs, averaged across 5 runs of all games in the *hard* difficulty level. We notice that actions requiring information about the goal of the game, like the `put` and `insert` actions, benefit more from attending to the commonsense graph; whereas actions aimed at exploring the environment and collecting objects, like the `go` and `take` actions, benefit more from the state representation.

As further qualitative analysis, we report an example of the most attended nodes and graphs from an excerpt of a game belonging to the *hard* difficulty level in Figure 4b. As noted above, the `take` and `go` actions rely more on the state graph, whereas the `insert` action relies on the commonsense graph. Among the nodes in these graphs, the entities that are finally mentioned in the action receive the highest attention score. This shows how our agent is able to transfer the bidirectional attention over graphs into specific game instances.

5 Conclusion

In this paper, we showed that in order to be sample-efficient in TBGs, agents must be able to jointly track the state of the game and relevant commonsense knowledge. We proposed a technique that models both forms of knowledge as graphs, and combines them using Bidirectional Knowledge-graph attEntion (BiKE). The resulting agent was found to be more sample-efficient than approaches that considered neither or only one of these graphs.

Broader Impact and Discussion of Ethics

While our model is not tuned for any specific real-world application, our method could be used in sensitive contexts such as legal or health-care settings, and it is essential that any work using our work undertake extensive quality-assurance and robustness testing before using it in their setting. The dataset used in our work does not contain any sensitive information to the best of our knowledge.

Replicability: As part of our contributions, we will release code for our agents and the game instances used for training and evaluating our models.

References

Ashutosh Adhikari, Xingdi Yuan, Marc-Alexandre Côté, Mikuláš Zelinka, Marc-Antoine Rondeau, Romain Laroche, Pascal Poupart, Jian Tang, Adam Trischler, and William L Hamilton. 2020. Learning dynamic knowledge graphs to generalize on text-based games. *arXiv preprint arXiv:2002.09127*.

Leonard Adolphs and Thomas Hofmann. 2019. Ledeepchef: Deep reinforcement learning agent for families of text-based games. *ArXiv*, abs/1909.01646.

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

Prithviraj Ammanabrolu and Mark Riedl. 2019. Playing text-adventure games with graph-based deep reinforcement learning. 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)*, pages 3557–3565.

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)*, pages 344–354.

Marc-Alexandre Côté, Ákos Kádár, Xingdi Yuan, Ben Kybartas, Tavian Barnes, Emery Fine, James Moore, Matthew Hausknecht, Layla El Asri, Mahmoud Adada, Wendy Tay, and Adam Trischler. 2018. Textworld: A learning environment for text-based games. *CoRR*, abs/1806.11532.

Ji He, Jianshu Chen, Xiaodong He, Jianfeng Gao, Lihong Li, Li Deng, and Mari Ostendorf. 2016. Deep reinforcement learning with a natural language action space. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1621–1630.

Brendan Juba. 2016. Integrated common sense learning and planning in pomdps. *The Journal of Machine Learning Research*, 17(1):3276–3312.

Hugo Liu and Push Singh. 2004. Conceptnet—a practical commonsense reasoning tool-kit. *BT technology journal*, 22(4):211–226.

Thang Luong, Hieu Pham, and Christopher D. Manning. 2015. Effective approaches to attention-based neural machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1412–1421, Lisbon, Portugal. Association for Computational Linguistics.

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 *International conference on machine learning*, pages 1928–1937.

Keerthiram Murugesan, Mattia Atzeni, Pavan Kapanipathi, Pushkar Shukla, Sadhana Kumaravel, Gerald Tesauro, Kartik Talamadupula, Mrinmaya Sachan, and Murray Campbell. 2021. Text-based RL Agents with Commonsense Knowledge: New Challenges, Environments and Baselines. In *The 35th AAAI Conference on Artificial Intelligence*.

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

Minjoon Seo, Aniruddha Kembhavi, Ali Farhadi, and Hannaneh Hajishirzi. 2016. Bidirectional attention flow for machine comprehension. *arXiv preprint arXiv:1611.01603*.

Matthijs TJ Spaan. 2012. Partially observable markov decision processes. In *Reinforcement Learning*, pages 387–414. Springer.

Robyn Speer, Joshua Chin, and Catherine Havasi. 2017. Conceptnet 5.5: An open multilingual graph of general knowledge. In *AAAI*, pages 4444–4451.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2018. Graph attention networks. In *International Conference on Learning Representations*.