Text-based RL Agents with Commonsense Knowledge: New Challenges, Environments and Baselines

Keerthiram Murugesan,[1, *] **Mattia Atzeni,**[1, 2] **Pavan Kapanipathi,**[1] **Pushkar Shukla,**[3] **Sadhana Kumaravel,**[1] **Gerald Tesauro,**[1] **Kartik Talamadupula,**[1] **Mrinmaya Sachan,**[4] **Murray Campbell** [1]
[1]IBM Research [2]EPFL [3]TTI Chicago [4]ETH Zurich

Abstract

Text-based games have emerged as an important test-bed for Reinforcement Learning (RL) research, requiring RL agents to combine grounded language understanding with sequential decision making. In this paper, we examine the problem of infusing RL agents with commonsense knowledge. Such knowledge would allow agents to efficiently act in the world by pruning out implausible actions, and to perform look-ahead planning to determine how current actions might affect future world states. We design a new text-based gaming environment called `TextWorld Commonsense` (`TWC`) for training and evaluating RL agents with a specific kind of commonsense knowledge about objects, their attributes, and affordances. We also introduce several baseline RL agents which track the sequential context and dynamically retrieve the relevant commonsense knowledge from ConceptNet. We show that agents which incorporate commonsense knowledge in `TWC` perform better, while acting more efficiently. We conduct user-studies to estimate human performance on `TWC` and show that there is ample room for future improvement.

Introduction

Over the years, simulation environments have been used extensively to drive advances in reinforcement learning (RL). A recent framework that has received much attention is TextWorld (`TW`) (Côté et al. 2018), where an agent must interact with an external environment to achieve a given goal using only the modality of text. TextWorld and similar text-based environments seek to bring advances in grounded language understanding to a sequential decision making setup.

While existing text-based games are valuable for RL research, they fail to test a key aspect of human intelligence: common sense. Humans capitalize on commonsense (background) knowledge about entities – properties, spatial relations, events, causes and effects, and other social conventions – while interacting with the world (Mccarthy 1960; Winograd 1972; Davis and Marcus 2015). Motivated by this, we propose a novel text-based environment called `TextWorld Commonsense` (or `TWC`), where the agent is expected to use commonsense knowledge stored in knowledge bases such as *ConceptNet* (Liu and Singh 2004; Speer, Chin, and Havasi 2017) to act efficiently. `TWC` is a sandbox environment similar to TextWorld where the agent has to clean up a house. Achieving goals in this environment requires commonsense knowledge about objects, their properties, locations, and affordances. Efficient use of commonsense knowledge would allow the agent to select correct and applicable actions at each step: i.e., improve sample efficiency by reducing exploration. Moreover, commonsense knowledge would help the agent to perform look-ahead planning and determine how current actions might affect future world states (Juba 2016). Fig 1 presents a running example from `TWC` that illustrates how the agent can leverage a commonsense knowledge base (KB).

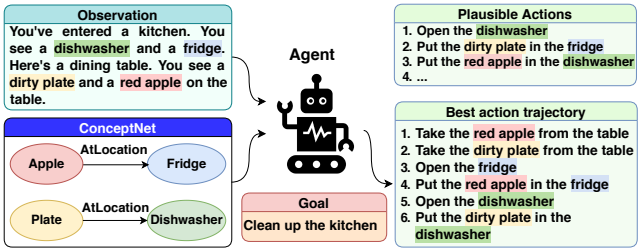

Figure 1: Illustration of a `TWC` game. The agent is given an initial observation (top left) and has to produce the list of actions (bottom right) that are necessary to achieve the goal (bottom center) using relevant commonsense knowledge from ConceptNet (bottom left).

Validating such environments is challenging, and requires: (1) verifying the information used in the games; (2) evaluating baseline agents that are capable of utilizing external commonsense knowledge against counterparts that do not; and (3) providing empirical evidence to show that the environment can drive future research. In this work, we address each of these by first performing human annotations to validate the correctness and completeness of the `TWC` environment. Next, we design a framework of agents that combine text-based agents with commonsense knowledge. The agents can dynamically retrieve relevant knowledge from a commonsense KB. Finally, based on human performance on

*Corresponding Author: keerthiram.murugesan@ibm.com. Code and data can be found at https://github.com/IBM/commonsense-rl.

the generated games and manual selection of commonsense knowledge, we discuss and justify the importance of such an environment in driving future research.

Contributions: The main contributions of this paper are the following: (1) we propose a novel environment called TWC to evaluate the use of commonsense knowledge by RL agents; (2) we introduce baselines that use commonsense knowledge from ConceptNet and show that common sense indeed helps in decision making; (3) while our model with common sense performs well, we show a pronounced gap in performance between automated agents and humans in the TWC environment. This substantiates our claim that TWC provides a challenging test-bed for RL agents and can act as a spur to further research in this area.

TextWorld Commonsense (TWC)

Existing text-based games (Adhikari et al. 2020; Côté et al. 2018) severely restrict the amount and variety of commonsense knowledge that an agent needs to know and exploit. Thus, in this paper, we create and present a new domain – TextWorld Commonsense (TWC) – by reusing the TextWorld (Côté et al. 2018) engine in order to generate text-based environments where RL agents need to effectively retrieve and use commonsense knowledge. Commonsense can be defined very broadly and in various ways (Fulda et al. 2017). In this paper, we mainly focus on commonsense knowledge that pertains to objects, their attributes, and affordances[1].

Constructing TWC

We built the TWC domain as a house clean-up environment where the agent is required to obtain knowledge about typical objects in the house, their properties, and expected location from a commonsense knowledge base. The environment is initialized with random placement of objects in various locations. The agent's high level goal is to tidy up the house by putting objects in their commonsense locations. This high level goal may consist of multiple sub-goals requiring commonsense knowledge. For example, for the sub-goal: *put the apple inside the refrigerator*, commonsense knowledge from ConceptNet such as (Apple → AtLocation → Refrigerator) can assist the agent.

Goal Sources: While our main objective was to create environments that require commonsense, we did not want to bias TWC towards any of the existing knowledge bases. We additionally wanted to rule out the possibility of data leaks in situations where both the environment as well as the external knowledge came from the same part of a specific commonsense knowledge base (KB) like ConceptNet. For the construction of the TWC goal instances, we picked sources of information that were orthogonal to existing commonsense KBs. Specifically, we used: (1) the *picture dictionary from 7ESL*[2]; (2) the *British Council's vocabulary learning page*[3]; (3) the *English At Home vocabulary learning* page[4]; and (4) *ESOL courses*[5]. We collected vocabulary terms from these sources and manually aggregated this content in order to build a dataset that lists several kinds of objects that are typically found in a house environment. For each object, the dataset specifies a list of plausible and coherent locations.

	Count	Examples
Rooms	8	*kitchen*, *backyard*
Supporters/Containers	56	*dining table*, *wardrobe*
Unique Objects	190	*plate*, *dress*
Total Objects	872	*dirty plate*, *clean red dress*
Total Entities	928	*dirty plate*, *dining table*

Table 1: Statistics on the number of entities, supporters/containers, and rooms in the TWC domain.

	Correctness	Completeness
Rated Commonsense	669	47
Rated NOT Commonsense	31	253

Table 2: Statistics from the human annotations to verify TWC

Instance Construction: A TWC instance is sampled from this dataset, which includes a configuration of 8 room types and a total of more than 900 entities (Table 1). The environment includes three main kinds of entities: objects, supporters, and containers. Objects are entities that can be carried by the agent, whereas supporters and containers are furniture where those objects can be placed. Let o represent the object or entity in the house; r represent the room that the entity is typically found in; and l represent the location inside that room where the entity is typically placed. In our running example, o:apple is an entity, l:refrigerator is the container, and r:kitchen is the room. Via a manual verification process (which we elucidate next in Section) we ensure that the associations between entities, supporters/containers, and rooms reflect commonsense. As shown in Table 1, we collected a total of 190 objects from the aforementioned resources. We further expanded this list by manually annotating the objects with qualifying properties, which are usually adjectives from a predefined set (e.g., a shirt may have a color and a specific texture). This allows increasing the cardinality of the total pool of objects for generating TWC environments to more than 800.

Verifying TWC

In order to ensure that TWC reflects commonsense knowledge, we set up two annotation tasks to verify the environment goals (i.e., goal triples of the form ⟨o, r, l⟩ , as defined in Section). The first task is meant to verify the correctness of the goals and evaluate whether the goal ⟨o, r, l⟩ triples make sense to humans. The second task is aimed at verifying completeness, i.e. that other triples in the environment do not make sense to humans.

[1] Gibson in his seminal work (Gibson 1978) refers to affordance as "properties of an object [...] that determine what actions a human can perform on them".

[2] https://7esl.com/picture-dictionary

[3] https://learnenglish.britishcouncil.org/vocabulary/beginner-to-pre-intermediate

[4] https://www.english-at-home.com/vocabulary

[5] https://www.esolcourses.com/topics/household-home.html

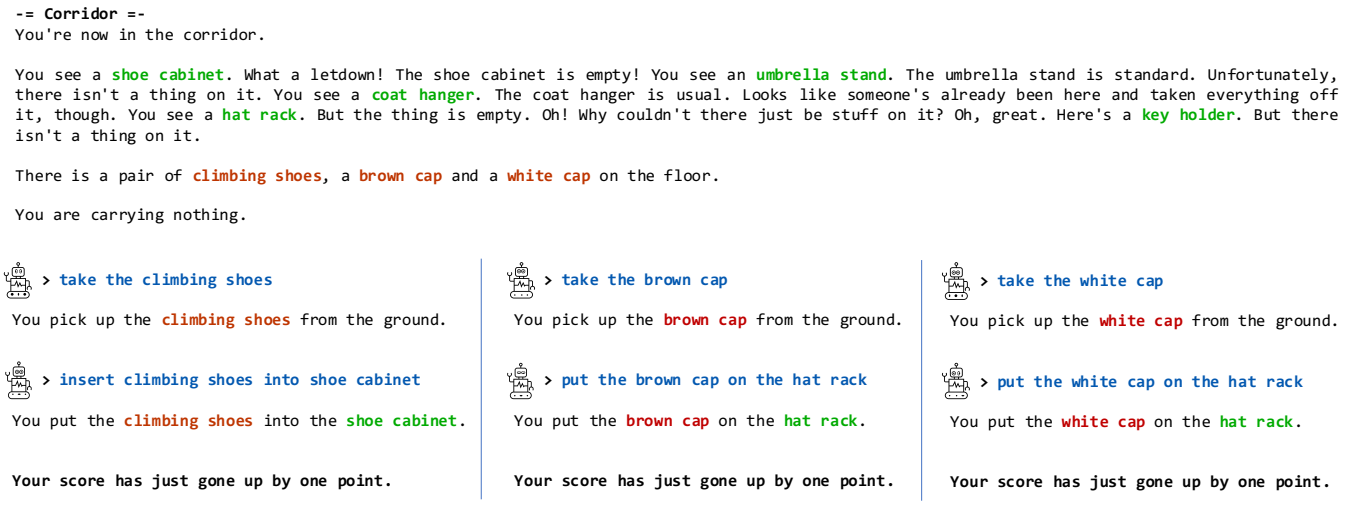

Figure 2: Sample game walkthrough for a game with *medium* difficulty level. Best viewed in colors. Highlights are not available to the agents and are shown for illustrative purpose only.

Verifying Correctness: To test the correctness of our environments, we asked our human annotators to determine whether they would consider a given room-location combination in the goal $\langle$o, r, l$\rangle$ to be a reasonable place for the object o. If so, the instance was labeled as positive, and as negative otherwise. We collected annotations from 10 annotators, across a total of 205 unique $\langle$o, r, l$\rangle$ triples. Each annotator labeled 70 of these triples, and each triple was assigned to at least 3 distinct annotators. The annotators were not given any other biasing information, and all annotators worked independently. We show the overall agreement of the annotators with TWC's goals in Table 2. The high agreement from the annotators demonstrates that the goal $\langle$o, r, l$\rangle$ triples reflect human commonsense knowledge.

Verifying Completeness: Similar to the above annotation exercise, we also asked human annotators to determine if a non-goal $\langle$o, r, l$\rangle$ triple made sense to them. In addition to the 70 triples mentioned above, each of the $M = 10$ annotators were asked to label as either positive or negative a set of 30 non-goal triples. In order to provide annotators with an informative set of non-goal $\langle$o, r, l$\rangle$ triples, we used GloVe (Pennington, Socher, and Manning 2014) to compute embeddings for each location in TWC. For a given object o, a non-goal location l' was then selected among those most similar to the goal location l, according to the cosine similarity between the embeddings of l and l'. As before, each non-goal triple was assigned to at least 3 annotators from a set that comprises a total of 97 triples. As we see in Table 2, the annotators seldom find a hypothesized non-goal $\langle$o, r, l$\rangle$ triple as commonsensical.

Annotator Reliability: For our overall annotation exercise, we can report inter-annotator agreement statistics, as the overall annotation is no longer imbalanced in terms of label marginals. We report a *Krippendorff's alpha* (Krippendorff 2018) $\alpha_\kappa = 0.74$. This number is over the accepted range for agreement and shows that our annotators have strong agreement when rating the triples.

	#Objects	#Objects to find	#Rooms
Easy	1	1	1
Medium	2, 3	1, 2, 3	1
Hard	6, 7	5, 6, 7	1, 2

Table 3: Specification of TWC games

Generating TWC Games

We used the TextWorld engine to build a set of text-based games where the goal is to tidy up a house by putting objects in the goal locations specified in the aforementioned TWC dataset. The games are grouped into three difficulty levels (easy, medium, and hard) depending on the total number of objects in the game, the number of objects that the agent needs to find (the remaining ones are already carried by the agent at the beginning of the game) and the number of rooms to explore. The values of these properties are randomly sampled from the ones listed in Table 3. For each difficulty level, we provide a training set and two test sets. The training sets were built out of $\frac{2}{3}$ of the unique objects reported in Table 1. For the first test set, we used the same set of objects as the training games. We call this set the *in* distribution test set. For the second test set, we employed the remaining $\frac{1}{3}$ objects to create the evaluation games. We call this set the *out* of distribution test set. This allows us to investigate not only the capability of the agents to generalize within the same distribution of the training data, but also their ability to achieve generalization to unseen entities. Fig 2 shows a game walkthrough for a specific game in the medium difficulty level.

Benchmarking Human Performance

To complete our benchmarking of the TWC domain, we conducted yet another human annotation task, focusing on the performance of human game-players. Such an experiment is essential to establishing the performance of human players, who are generally regarded as proficient at exploiting com-

monsense knowledge. We set up an interactive interface to TWC via a Jupyter notebook, which was then used by players to interact with the same games that we evaluated all the other RL agents on. We recorded all moves (steps) made by players, as well as the reward collected. At each step, the players were shown the current context of the game in text format, and given a drop-down box with the full list of possible actions. Once the player picked an action, it was executed; and this process repeated until all possible goals in the game had been accomplished. A total of 16 annotators played 104 instances of TWC games, spread across the *easy*, *medium*, and *hard* levels. Each difficulty level had 5 games, each from the train and test distributions, for a total of 30 unique games. Each unique game was annotated by a minimum of 3 annotators. The results are presented in Table 4, along with the experimental results in Section , to allow for direct comparison with the TWC agents.

TWC Agents

Text-based games can be seen as partially observable Markov decision processes (POMDP) (Kaelbling, Littman, and Cassandra 1998) where the system dynamics are determined by an MDP, but the agent cannot directly observe the underlying state. The agent receives a reward at every time step and its goal is to maximize the expected discounted sum of rewards. The TWC games allow the agent to perceive and interact with the environment via text. Thus, the observation at time step t, o_t, is presented as a sequence of tokens ($o_t = \{o_t^1, \dots o_t^N\}$). Similarly, each action a is also denoted as a sequence of tokens $\{a^1, \dots, a^M\}$. The goal of this project is to test RL agents with commonsense. Hence, the agents also have access to a commonsense knowledge base; and are allowed to use it while selecting actions. To model TWC, we design a framework that can: (a) learn representations of various actions; (b) learn from sequential context; (c) dynamically retrieve the relevant commonsense knowledge; (d) integrate the retrieved commonsense knowledge with the context; and (e) predict next action. A block diagram of the framework is shown in Fig 3. We describe the various components of our framework below.

Action and Observation Encoder

We learn representations of observations and actions by feeding them to a recurrent network. Given the current observation o_t, we use pre-trained word embeddings to represent o_t as a sequence of d-dimensional vectors $\mathbf{x}_t^1, \dots, \mathbf{x}_t^N$, where each $\mathbf{x}_t^k \in \mathbb{R}^d$ is the word embedding of the k-th observed token o_t^k, $k = 1, \dots, N$. Then, a (bidirectional) GRU encoder (Cho et al. 2014) is used to process the sequence $\mathbf{x}_t^1, \dots, \mathbf{x}_t^N$ to get the representation of the current observation: $\mathbf{o}_t = \mathbf{h}_t^N$, where $\mathbf{h}_t^k = GRU(\mathbf{h}_t^{k-1}, \mathbf{x}_t^k)$, for $k = 1, \dots, N$. In a similar way, given the set A_t of admissible actions at time step t, we learn representations of each action $a \in A_t$.

Context Encoder

A key challenge for our RL agent is in modeling context, i.e. the history of observations. We model the context using another recurrent encoder over the observation representations

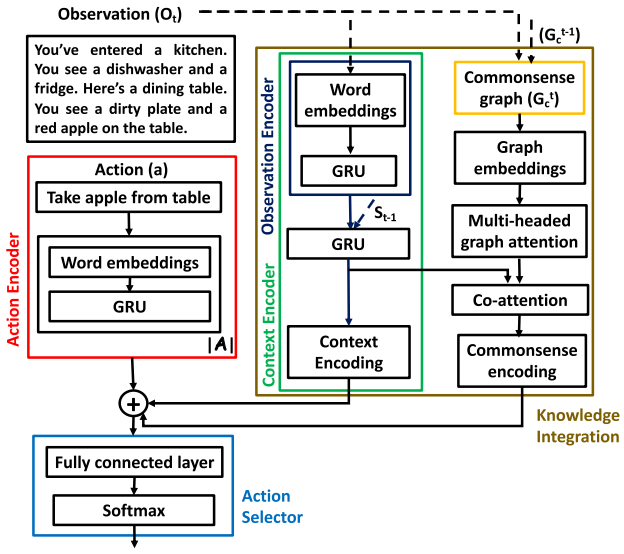

Figure 3: Overview of our framework's decision making at any given time step. The framework comprises of the following components (visually shown in color): (a) *action encoder* which encodes all admissible actions $a \in \mathscr{A}$, (b) *observation encoder* which encodes the observation o_t, (c) *context encoder*, which encodes the dynamic context C_t, (d) a dynamic *common sense subgraph* of ConceptNet G_C^t extracted by the agent, (e) a *knowledge integration* component, which combines the information from textual observations and the extracted common sense subgraph, and (f) an *action selection* module. $\oplus$ denotes the concatenation operator.

$\mathbf{o}_t$. We use a GRU network to encode the sequence of previous observations up to o_t into a vector $\mathbf{s}_t = GRU(\mathbf{s}_{t-1}, \mathbf{o}_t)$. We refer to s_t as the state vector, or the context encoding. The context encoding will be used in addition to the commonsense knowledge in the final action prediction.

Dynamic Commonsense Subgraph

Our model retrieves commonsense knowledge from ConceptNet in the form of a graph. The graph G_C^t is updated dynamically at each time step t. G_C^t is constructed by mapping the textual observation o_t at time t to ConceptNet and combining it with the graph at previous time step G_C^{t-1}. We used *spaCy* (https://spacy.io) to extract noun chunks, and then performed a max sub-string match with all the concepts in ConceptNet. This results in a set of entities e_t for the observation o_t at time t. We then combine the concepts from G_C^{t-1} and e_t to get E_t. E_t consists of all the concepts observed by the agent until time step t, including the description of the room, the current observation, and the objects in the inventory. Given E_t, we describe three different techniques that automatically extract the commonsense graph G_t from external knowledge.

(1) Direct Connections (DC): This is the baseline approach to construct G_C^t. We fetch direct links between each of the concepts in E_t from ConceptNet.

(2) Contextual Direct Connections (CDC): Since the goal

of the agent is to clean up the house by putting `objects` into its appropriate `containers` such as `apple` ⇒ `refrigerator`, we hypothesize that adding links only between objects and containers may benefit the agent instead of links between all concepts as done by *Direct Connections*, as we might overwhelm the agent with noise. To accomplish this goal, we split the entities E_t into objects and containers. Since we know the entities from the inventory in E_t constitutes objects, no explicit labelling is needed as we consider the remaining entities as containers. We retain only the edges between objects and containers from ConceptNet.

(3) Neighborhood (`NG`): Previous techniques only focus on connecting links between observed concepts E_t from external knowledge. In addition to the direct relations, it may be beneficial to include concepts from external knowledge that are related to E_t, but have not been directly observed from the game. Therefore, for each concept in E_t, we include all its neighboring concepts and associated links.

Knowledge Integration

We enhance the text-based RL agent by allowing it to jointly contextualize information from both the commonsense subgraph and the observation representation. We call this step knowledge integration. We encode the commonsense graph using a graph encoder followed by a co-attention layer.

Graph encoder: The graph G_C^t is encoded as follows: First, we use pretrained KG embeddings (Numberbatch) to map the set of nodes $\mathscr{V}_t$ to a feature matrix $[\mathbf{e}_t^1, \ldots, \mathbf{e}_t^{|\mathscr{V}_t|}] \in \mathbb{R}^{f \times |\mathscr{V}_*^t|}$. Here, $\mathbf{e}_t^i \in \mathbb{R}^f$ is the (averaged) embedding of words in node $i \in \mathscr{V}_*^t$. Following (Lu et al. 2017), we also add a *sentinel* vector to allow the attention modules to not attend to any specific nodes in the subgraph. These node embeddings are updated at each time step by message passing between the nodes of G_c^t with Graph Attention Networks (GATs) (Veličković et al. 2018) to get $\{\mathbf{z}_t^1, \mathbf{z}_t^2 \cdots \mathbf{z}_t^{|\mathscr{V}_t|}\}$ using multi-head graph attention, resulting in a final graph representation that better captures the conceptual relations between the nodes in the subgraph.

Co-Attention: In order to combine the observational context and the retrieved commonsense graph, we consider a bidirectional attention flow layer between these representations to re-contextualize the graph for the current state of the game (Seo et al. 2016; Yu et al. 2018). Similar to (Yu et al. 2018), we compute a similarity matrix $S \in \mathbb{R}^{N \times |\mathscr{V}_C^t|}$ between the context and entities in the extracted common sense subgraph using a trilinear function. In particular, the similarity between i^{th} token's context encoding $\mathbf{h}_t^i$ and j^{th} node encoding $\mathbf{z}_t^j$ in the commonsense subgraph is computed as: $S_{ij} = \mathbf{W}_0^T[\mathbf{h}_t^i; \mathbf{z}_t^j; \mathbf{h}_t^i \circ \mathbf{z}_t^j]$ where $\circ$ denotes element-wise product, ; denotes concatenation and $\mathbf{W}_0$ is a learnable parameter. We use the softmax function to normalize the rows (columns) of S and get the similarity function for the common-sense knowledge graph $\bar{S}_G$ (context representation $\bar{S}_O$). The commonsense-to-context attention is calculated as $A = \bar{S}_G^\top \cdot O$ and the context-to-common sense attention is calculated as $B = \bar{S}_G^\top \bar{S}_O \cdot G$, where $G = [\mathbf{z}_t^1, \mathbf{z}_t^2, \cdots \mathbf{z}_t^{|\mathscr{V}_C^t|}]^\top$ and $O = [\mathbf{h}_t^1, \mathbf{h}_t^2 \cdots \mathbf{h}_t^N]^\top$ are the commonsense graph and observation encodings. The attention vectors are then combined together and the final graph encoding vectors $\mathbf{G}$ are calculated as $\mathbf{W}^\top[G; \mathbf{A}; G \circ \mathbf{A}; G \circ \mathbf{B}]$ where W is the learnable parameter. Finally, we get the commonsense graph encoding $\mathbf{g}_i^t$ for each action $a_i \in A_t$ by applying a general attention over the nodes using the state vector and the action encoding $[\mathbf{s}_t; \mathbf{a}_i^t]$ (Luong, Pham, and Manning 2015). The attention score for each node is computed as $\alpha_i = [\mathbf{s}_t; \mathbf{a}_i^t]\mathbf{W}_g\mathbf{G}$, and the commonsense graph encoding for action $\mathbf{a}_i^t$ is given as $\mathbf{g}_i^t = \alpha_i^\top G$.

Action Selection

The action score for each action $\hat{a}_i^t$ is computed based on the context encoding $\mathbf{s}_t$, the commonsense graph encoding $\mathbf{g}_i^t$ and the action encoding $\mathbf{a}_i^t$. We concatenate these encoding vectors into a single vector $\mathbf{r}_i^t = [\mathbf{s}_t; \mathbf{g}_i^t; \mathbf{a}_i^t]$. Then, we compute probability score for each action $a_i \in A_t$ as $\mathbf{p}_t = softmax(W_1 \cdot ReLU(W_2 \cdot \mathbf{r}_t + \mathbf{b}_2) + \mathbf{b}_1)$; where $W_1, W_2, \mathbf{b}_1$, and $\mathbf{b}_2$ are learnable parameters of the model. The final action chosen by the agent is then given by the one with the maximum probability score, namely $\hat{a}_t = \arg\max_i p_{t,i}$.

Experiments

In this section, we report the results of our experiments on the `TWC` games. Given that the quality (correctness and completeness) of `TWC` has already been evaluated (c.f. Section), these experiments primarily focus on showing that: (1) agents that utilize commonsense knowledge can achieve better performance on `TWC` than their text-based counterparts; (2) `TWC` can aid research in the use of commonsense knowledge because of the gap between human performance and the commonsense knowledge agents.

Experimental Setup: We measure the performance of the various agents using: (1) the normalized score (score achieved ÷ maximum achievable score); and (2) the number of steps taken. Each agent is trained for 100 episodes and the results are averaged over 10 runs. Following one of the winning strategies in the *FirstTextWorld Competition* (Adolphs and Hofmann 2019), we use the Advantage Actor-Critic framework (Mnih et al. 2016) to train the agents using reward signals from the training games.

RL Agents in `TWC`

We evaluate our framework on the `TWC` cleanup games (as described in Section). For comparison, we consider a random agent that randomly picks an action at each time step. We consider two types of experiment settings based on the type of information available to the RL agents: (1) `Text-based` RL agents have access to the textual description (observation) of the current state of the game provided by the `TWC` environment; and (2) `Commonsense-based` RL agents have access to both the observation and ConceptNet.

Text-only Baseline Agents: As baselines, we picked various SOTA text-based agents that utilize observation only: (1) **LM-NSP** uses language models such as *BERT* (Devlin et al. 2019) and *GPT2* (Radford et al. 2019) with the observation and the action pair as a Next Sentence Prediction (NSP)

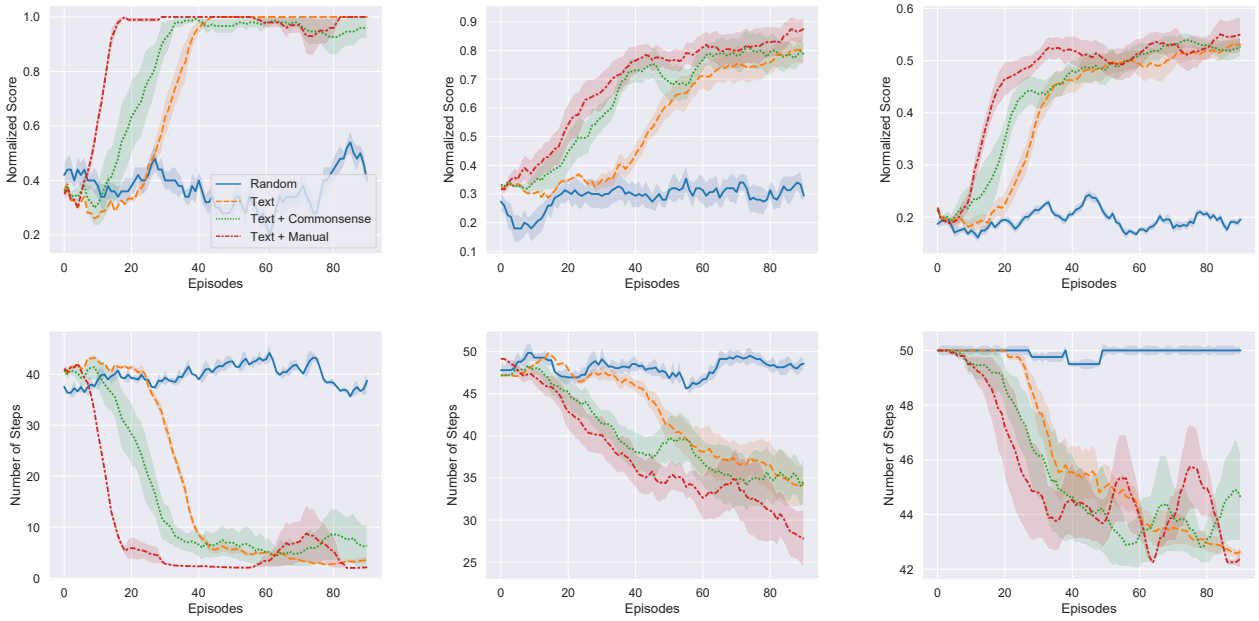

Figure 4: Performance evaluation (showing mean and standard deviation averaged over 10 runs) for the three difficulty levels: Easy (left), Medium (middle), Hard (right) using normalized score and the number of steps taken.

task; (2) **LSTM-A2C** (Narasimhan, Kulkarni, and Barzilay 2015) uses the observed text to select the next action; (3) **DRRN** (He et al. 2016) utilizes the relevance between the observation and action spaces for better convergence; and (4) **KG-A2C** (Ammanabrolu and Hausknecht 2020) uses knowledge of the game environment generated from the observation to guide the agent's exploration. For these baselines, we use GloVe (Pennington, Socher, and Manning 2014) embeddings for text.

The results on these baselines are reported in Table 4. For each difficulty level, we report: the agents' performance; the optimal number of steps to solve the game[6]; and the human performance. The performance of GPT2-NSP and BERT-NSP shows that even powerful pretrained models if not tuned to this task have difficulty in these commonsense RL games, as they do not capture commonsense relationships between entities. Baselines such as LSTM-A2C, DRRN, and KG-A2C have a competitive advantage over the LM-NSP baselines, as they effectively adapt to the sequential interaction with the environment to improve performance. Among these baselines, DRRN and KG-A2C perform better than LSTM-A2C as they utilize the structure of the state and action spaces for efficient exploration of the environment.

Commonsense-based agents: We introduce commonsense knowledge in two ways. The first is (Text + Numberbatch) by replacing GloVe embeddings in the LSTM-A2C agent with Numberbatch (*Nb*) embeddings (Speer, Chin, and Havasi 2017) which were trained on text and ConceptNet. This is the naive approach to augment text information with commonsense knowledge. The results in Table 4 show that introducing *Nb* embeddings allows achieving a noticeable gain (an average of 3 steps in easy and 7 steps in medium level games) over GloVe embeddings.

In order to explicitly use commonsense knowledge, we experiment with the three different mechanisms outlined in Section for retrieving relevant information from ConceptNet: (`DC`, `CDC` and `NG`). These methods retrieve both the concepts and structure in the relevant sub-graphs from ConceptNet, which are leveraged by our co-attention mechanism (Section). The comparison of the agents' performance with different retrieving mechanisms is shown in Fig 5. The results show that `CDC` performs the best among other mechanisms, particularly compared to `DC`. Unlike `DC` that includes all the links between observed concepts from ConceptNet, `CDC` restricts links to those between observed objects and *containers*. This selection of relevant links from ConceptNet improves the performance of the agent.

Given that `CDC` performs best, we compare results on text-based models with `CDC`-augmented commonsense knowledge to other baselines. Table 4 shows results for text-based agents initialized with GloVe or *Nb* embeddings, and augmented with commonsense knowledge. We see that the commonsense-based RL agents perform better than text-based RL agents in the easy and medium level games. This is not surprising, as these instances mostly involve picking an object and placing it in a container in the same room. Both the text-based and commonsense RL agents struggle in the hard level, as these games have more than one room and multiple objects and containers. We also notice that the average number of steps taken by the commonsense-based

[6]The optimal number of steps were computed by considering the objects already in the agent's possession, the number of objects to "put" (goals), and the number of rooms in the instance.

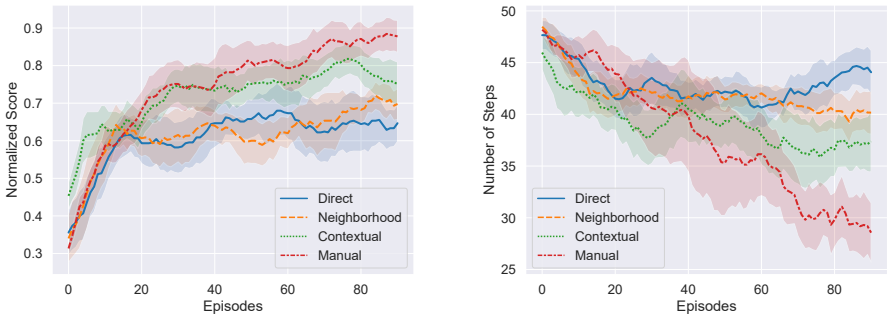

Figure 5: Performance for the medium level (train-set) games (showing mean and standard deviation averaged over 3 runs) with the different techniques for the commonsense sub-graph extraction.

RL agents are noticeably lower than the other agents as it efficiently uses commonsense knowledge to rule out implausible actions. This proves that TWC is a promising test-bed where commonsense knowledge helps.

Our results show that TWC still has much room for improvement in terms of retrieving and combining knowledge with observations and feedback from the environment in a sample-efficient manner. As a starting point for showing that there is headroom, we switched the retrieval mechanism to manually selected information from ConceptNet. We manually retrieved the relevant commonsense knowledge by extracting the commonsensical paths between entities in ConceptNet, corresponding to objects in the TWC games and their goal locations. The manual subgraph includes all the relevant shortest paths between an object and its location, within a 2-hop neighborhood expansion of both nodes. Since the extracted subgraph can be very large even for the easy games, further pruning was performed to remove noise. We emphasize that the manual annotation can be error-prone or result in manual subgraphs that lack potentially useful information. Thus, the manual graphs should not be taken as a gold standard. However, we are exploring other manual retrieval process to understand if better commonsense retrieval approaches can bring improvements in the future. In Table 4, agents that are augmented with the manual graph perform better than the other automated retrieval mechanisms (average reduction of $2-5$ steps on easy and medium). Fig 4 shows training curves for the Text-only, Text+Commonsense and Text+Manual agents on the three difficulty levels. We notice that infusing commonsense knowledge allows achieving faster convergence both in terms of the number of steps taken by the agents and the final score. We found that the extracted manual subgraphs is not perfect as can be seen in the training curves for medium and hard levels.

Human Performance on TWC: We also present the results of human performance in TWC (outlined in Section). The **O** and **H** columns in Table 4 (two per condition) present these results. A quick comparison of these numbers reveals two major results: (1) human performance **H** is very close to the optimal number of steps **O** in all 3 conditions; and (2) there is significant headroom between **H** and all of the other agents in the table, including the ones with the manual graph. This confirms that there is still much progress to be made in retrieving and encoding the commonsense knowledge effectively to solve such problems; and that TWC can spur further research.

Generalization

Table 4 reports the results both for test games that belong to the same distribution used at training time (**IN**), and games that were generated from a different set of entities (**OUT**). We see a similar trend on both these settings. The commonsense-enhanced agent outperforms the text-only agent in all cases. However, all agents including those that utilize commonsense knowledge show similar drop in performance from **IN** to **OUT** distribution. This is in contrast to the use of the knowledge graphs in other NLP tasks such as textual entailment where knowledge graphs have shown to be robust to changes in the underlying (training and testing) environment (Kapanipathi et al. 2020; Chen et al. 2018). The task of designing knowledge-enabled agents that are robust to such changes is another open challenge for the community that can be evaluated by TWC.

Results Summary: Our results establish that TWC is an environment where agents augmented with commonsense knowledge show better performance than their text-based counterparts. Based on the experiments with manually retrieved sub-graphs, optimal steps, and the human performance numbers, we show that TWC has enough headroom for future research efforts to: (1) retrieve more relevant commonsense knowledge for KBs; and (2) for new agents/techniques to exploit such knowledge.

Related Work

RL Environments and TextWorld: Games are a rich domain for studying grounded language and how information from text can be utilized in control. Recent work has explored text-based RL games to learn strategies for *CivII* (Branavan, Silver, and Barzilay 2012), multi-user dungeon games (Narasimhan, Kulkarni, and Barzilay 2015), etc. Our work builds on the TextWorld (Côté et al. 2018) sandbox learning environment. Since its introduction, there has been a large body of work devoted to improving performance on

		Easy				Medium				Hard			
		O	H	#Steps	Norm. Score	O	H	#Steps	Norm. Score	O	H	#Steps	Norm. Score
IN	**GPT2-NSP**	2.00 ± 0.00	2.12 ± 0.49	30.36 ± 0.00	0.64 ± 0.00	3.60 ± 0.55	5.33 ± 2.06	42.12 ± 0.00	0.70 ± 0.00	15.00 ± 2.00	15.00 ± 3.29	50.00 ± 0.00	0.36 ± 0.00
	BERT-NSP			25.20 ± 0.00	0.76 ± 0.00			34.72 ± 0.00	0.88 ± 0.00			50.00 ± 0.00	0.52 ± 0.00
	LSTM-A2C			17.59 ± 3.11	0.86 ± 0.04			37.99 ± 6.03	0.74 ± 0.11			49.21 ± 0.58	0.54 ± 0.04
	DRRN			18.88 ± 2.69	0.81 ± 0.08			33.41 ± 2.81	0.73 ± 0.06			46.20 ± 4.86	0.44 ± 0.01
	KG-A2C			17.65 ± 3.62	0.85 ± 0.07			37.18 ± 4.86	0.72 ± 0.07			49.36 ± 7.50	0.46 ± 0.10
	Text												
	+Commonsense			14.18 ± 6.47	0.89 ± 0.10			34.67 ± 6.65	0.78 ± 0.07			48.45 ± 2.50	0.51 ± 0.10
	+Manual			13.70 ± 1.85	0.92 ± 0.03			29.26 ± 0.94	0.88 ± 0.03			46.43 ± 3.67	0.54 ± 0.04
	+Numberbatch			11.79 ± 3.04	0.96 ± 0.03			27.10 ± 5.06	0.85 ± 0.06			44.22 ± 4.86	0.57 ± 0.00
	+Nb+Commonsense			14.43 ± 3.08	0.93 ± 0.06			25.11 ± 2.33	0.87 ± 0.04			43.27 ± 0.70	0.45 ± 0.00
	+Nb+Manual			13.37 ± 5.63	0.92 ± 0.07			23.51 ± 1.28	0.91 ± 0.06			42.87 ± 0.65	0.52 ± 0.01
OUT	**GPT2-NSP**	2.00 ± 0.00	2.24 ± 0.75	40.28 ± 0.00	0.46 ± 0.00	4.40 ± 1.14	4.40 ± 1.85	44.96 ± 0.00	0.38 ± 0.00	14.60 ± 2.67	17.67 ± 3.31	50.00 ± 0.00	0.14 ± 0.00
	BERT-NSP			24.76 ± 0.00	0.72 ± 0.00			41.12 ± 0.00	0.55 ± 0.00			50.00 ± 0.00	0.27 ± 0.00
	LSTM-A2C			19.89 ± 1.86	0.79 ± 0.01			43.70 ± 5.52	0.52 ± 0.18			50.00 ± 0.00	0.27 ± 0.01
	DRRN			19.49 ± 4.89	0.84 ± 0.08			40.49 ± 4.41	0.56 ± 0.07			50.00 ± 0.00	0.18 ± 0.10
	KG-A2C			18.00 ± 3.24	0.87 ± 0.05			43.08 ± 4.13	0.54 ± 0.17			49.96 ± 0.00	0.22 ± 0.00
	Text												
	+Commonsense			19.14 ± 3.32	0.83 ± 0.07			41.01 ± 6.97	0.56 ± 0.13			49.99 ± 0.01	0.28 ± 0.05
	+Manual			16.86 ± 2.26	0.89 ± 0.04			39.95 ± 2.46	0.71 ± 0.06			49.97 ± 0.04	0.26 ± 0.11
	+Numberbatch			19.77 ± 2.50	0.81 ± 0.15			34.54 ± 2.89	0.80 ± 0.04			49.95 ± 0.08	0.29 ± 0.02
	+Nb+Commonsense			20.84 ± 1.13	0.83 ± 0.03			33.43 ± 2.11	0.71 ± 0.09			50.00 ± 0.00	0.25 ± 0.01
	+Nb+Manual			18.24 ± 4.63	0.83 ± 0.09			30.12 ± 4.62	0.84 ± 0.03			49.99 ± 0.02	0.22 ± 0.05

Table 4: Generalization results for within distribution (*IN*) and out-of-distribution (*OUT*) games. *O* represents the optimal #steps needed to accomplish the goals. *H* represents human-level performance. All agents were restricted to a max of 50 steps.

this benchmark. A recent line of work on TextWorld learns symbolic representations of the agent's belief. Notably, Ammanabrolu and Riedl (2019) proposed *KG-DQN* and Adhikari et al. (2020) proposed *GATA*. Both approaches represent the game state as a belief graph. This graph is used to prune the action space, enabling efficient exploration, in a different way from our work which uses common sense. The *LeDeepChef* system (Adolphs and Hofmann 2019) is also related to our work. They achieve transfer by additionally supervising the model with a list of the most common food items in `FreeBase` (Bollacker et al. 2008), allowing their agent to generalize to hitherto unseen recipes and ingredients. Zahavy et al. (2018) propose the Action-Elimination Deep Q-Network (AE-DQN), which learns to predict invalid actions in the text-adventure game *Zork*. This network allows the model to efficiently handle the large action space. The use of commonsense knowledge in our work potentially has the same effect of down-weighting implausible actions.

External Knowledge for Efficient RL: There have been few attempts on adding prior or external knowledge to RL approaches. Notably, Garnelo, Arulkumaran, and Shanahan (2016) proposed *Deep Symbolic RL*, which combines aspects of symbolic AI with neural networks and RL as a way to introduce commonsense priors. There has also been work on *policy transfer* (Bianchi et al. 2015), which studies how knowledge acquired in one environment can be re-used in another one; and *experience replay* (Wang et al. 2016; Lin 1992, 1993) which studies how an agent's previous experiences can be stored and then later reused. In this paper, we use commonsense knowledge as a way to improve sample efficiency in text-based RL agents. To the best of our knowledge, there is no prior work that *practically* explores how commonsense can be used to make RL agents more efficient. The most relevant prior work is by Martin, Sood, and Riedl (2018), who use commonsense rules to build agents that can play tabletop role-playing games. However, unlike our work, the commonsense rules in this work are manually engineered.

Leveraging Commonsense: Recently, there has been a lot of work in NLP to utilize commonsense for QA, NLI, etc. (Sap et al. 2019; Talmor et al. 2018). Many of these approaches seek to effectively utilize ConceptNet by reducing the noise retrieved from it (Lin et al. 2019; Kapanipathi et al. 2020). This is also a key challenge in `TWC`.

Conclusion

We created a novel environment (`TWC`) to evaluate the performance on RL agents on text-based games requiring commonsense knowledge. We introduced a framework of agents which tracks the state of the world; uses the sequential context to dynamically retrieve relevant commonsense knowledge from a knowledge graph; and learns to combine the two different modalities. Our agents equipped with common sense achieve their goals with greater efficiency and less exploration when compared to a text-only model, thus showing the value of our new environments and models. Therefore, we believe that our `TWC` environment provides interesting challenges and can be effectively used to fuel further research in this area.

Reproducibility

To ensure the wide and unrestricted usage of the `TWC` environment, we release the `TWC` environment (with *anonymized* human annotations), code to generate text-based games, and the sample agents used in this paper: https://github.com/IBM/commonsense-rl.

References

Adhikari, A.; Yuan, X.; Côté, M.-A.; Zelinka, M.; Rondeau, M.-A.; Laroche, R.; Poupart, P.; Tang, J.; Trischler, A.; and Hamilton, W. L. 2020. Learning Dynamic Knowledge Graphs to Generalize on Text-Based Games. *arXiv preprint arXiv:2002.09127* .

Adolphs, L.; and Hofmann, T. 2019. LeDeepChef: Deep Reinforcement Learning Agent for Families of Text-Based Games. *ArXiv* abs/1909.01646.

Ammanabrolu, P.; and Hausknecht, M. 2020. Graph Constrained Reinforcement Learning for Natural Language Action Spaces. In *International Conference on Learning Representations*. URL https://openreview.net/forum?id=B1x6w0EtwH.

Ammanabrolu, P.; and Riedl, M. 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)*, 3557–3565.

Bianchi, R. A.; Celiberto Jr, L. A.; Santos, P. E.; Matsuura, J. P.; and de Mantaras, R. L. 2015. Transferring knowledge as heuristics in reinforcement learning: A case-based approach. *Artificial Intelligence* 226: 102–121.

Bollacker, K.; Evans, C.; Paritosh, P.; Sturge, T.; and Taylor, J. 2008. Freebase: a collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, 1247–1250. AcM.

Branavan, S.; Silver, D.; and Barzilay, R. 2012. Learning to win by reading manuals in a monte-carlo framework. *Journal of Artificial Intelligence Research* 43: 661–704.

Chen, Q.; Zhu, X.; Ling, Z.-H.; Inkpen, D.; and Wei, S. 2018. Neural natural language inference models enhanced with external knowledge. In *Proceedings of ACL 2018*.

Cho, K.; van Merrienboer, B.; Gülçehre, Ç.; Bahdanau, D.; Bougares, F.; Schwenk, H.; and Bengio, Y. 2014. Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation. In Moschitti, A.; Pang, B.; and Daelemans, W., eds., *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*, 1724–1734. ACL. doi:10.3115/v1/d14-1179. URL https://doi.org/10.3115/v1/d14-1179.

Côté, M.-A.; Kádár, A.; Yuan, X.; Kybartas, B.; Barnes, T.; Fine, E.; Moore, J.; Hausknecht, M.; Asri, L. E.; Adada, M.; Tay, W.; and Trischler, A. 2018. TextWorld: A Learning Environment for Text-based Games. *CoRR* abs/1806.11532.

Davis, E.; and Marcus, G. 2015. Commonsense reasoning and commonsense knowledge in artificial intelligence. *Communications of the ACM* 58(9): 92–103.

Devlin, J.; Chang, M.; Lee, K.; and Toutanova, K. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In Burstein, J.; Doran, C.; and Solorio, T., eds., *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, 4171–4186. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/N19-1423/.

Fulda, N.; Ricks, D.; Murdoch, B.; and Wingate, D. 2017. What Can You Do with a Rock? Affordance Extraction Viaword Embeddings. IJCAI'17. AAAI Press. ISBN 9780999241103.

Garnelo, M.; Arulkumaran, K.; and Shanahan, M. 2016. Towards deep symbolic reinforcement learning. *arXiv preprint arXiv:1609.05518* .

Gibson, J. J. 1978. The ecological approach to the visual perception of pictures. *Leonardo* 11(3): 227–235.

He, J.; Chen, J.; He, X.; Gao, J.; Li, L.; Deng, L.; and Ostendorf, M. 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)*, 1621–1630.

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

Kaelbling, L. P.; Littman, M. L.; and Cassandra, A. R. 1998. Planning and acting in partially observable stochastic domains. *Artificial intelligence* 101(1-2): 99–134.

Kapanipathi, P.; Thost, V.; Patel, S. S.; Whitehead, S.; Abdelaziz, I.; Balakrishnan, A.; Chang, M.; Fadnis, K.; Gunasekara, C.; Makni, B.; Mattei, N.; Talamadupula, K.; and Fokoue, A. 2020. Infusing Knowledge into the Textual Entailment Task Using Graph Convolutional Networks. *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)* .

Krippendorff, K. 2018. *Content analysis: An introduction to its methodology*. Sage publications.

Lin, B. Y.; Chen, X.; Chen, J.; and Ren, X. 2019. Kagnet: Knowledge-aware graph networks for commonsense reasoning. *arXiv preprint arXiv:1909.02151* .

Lin, L.-J. 1992. Self-improving reactive agents based on reinforcement learning, planning and teaching. *Machine learning* 8(3-4): 293–321.

Lin, L.-J. 1993. Reinforcement learning for robots using neural networks. Technical report, Carnegie-Mellon Univ Pittsburgh PA School of Computer Science.

Liu, H.; and Singh, P. 2004. ConceptNet—a practical commonsense reasoning tool-kit. *BT technology journal* 22(4): 211–226.

Lu, J.; Xiong, C.; Parikh, D.; and Socher, R. 2017. Knowing when to look: Adaptive attention via a visual sentinel for image captioning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 375–383.

Luong, T.; Pham, H.; and Manning, C. D. 2015. Effective Approaches to Attention-based Neural Machine Translation. In *Proceedings of the 2015 Conference on Empirical*

Methods in Natural Language Processing, 1412–1421. Lisbon, Portugal: Association for Computational Linguistics. doi:10.18653/v1/D15-1166. URL https://www.aclweb.org/anthology/D15-1166.

Martin, L. J.; Sood, S.; and Riedl, M. 2018. Dungeons and DQNs: Toward Reinforcement Learning Agents that Play Tabletop Roleplaying Games. In Wu, H.; Si, M.; and Jhala, A., eds., *Proceedings of the Joint Workshop on Intelligent Narrative Technologies and Workshop on Intelligent Cinematography and Editing co-located with 14th AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment, INTWICED@AIIDE 2018, Edmonton, Alberta, Canada, November 13-14, 2018*, volume 2321 of *CEUR Workshop Proceedings*. CEUR-WS.org. URL http://ceur-ws.org/Vol-2321/paper4.pdf.

Mccarthy, J. W. 1960. Programs with common sense. In *Teddington Conference on the Mechanization of Thought Processes*.

Mnih, V.; Badia, A. P.; Mirza, M.; Graves, A.; Lillicrap, T.; Harley, T.; Silver, D.; and Kavukcuoglu, K. 2016. Asynchronous methods for deep reinforcement learning. In *International conference on machine learning*, 1928–1937.

Narasimhan, K.; Kulkarni, T.; and Barzilay, R. 2015. Language understanding for text-based games using deep reinforcement learning. *arXiv preprint arXiv:1506.08941* .

Pennington, J.; Socher, R.; and Manning, C. D. 2014. Glove: Global Vectors for Word Representation. In Moschitti, A.; Pang, B.; and Daelemans, W., eds., *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*, 1532–1543. ACL. doi:10.3115/v1/d14-1162. URL https://doi.org/10.3115/v1/d14-1162.

Radford, A.; Wu, J.; Child, R.; Luan, D.; Amodei, D.; and Sutskever, I. 2019. Language Models are Unsupervised Multitask Learners. *OpenAI Blog* .

Sap, M.; Le Bras, R.; Allaway, E.; Bhagavatula, C.; Lourie, N.; Rashkin, H.; Roof, B.; Smith, N. A.; and Choi, Y. 2019. Atomic: An atlas of machine commonsense for if-then reasoning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, 3027–3035.

Seo, M.; Kembhavi, A.; Farhadi, A.; and Hajishirzi, H. 2016. Bidirectional attention flow for machine comprehension. *arXiv preprint arXiv:1611.01603* .

Speer, R.; Chin, J.; and Havasi, C. 2017. ConceptNet 5.5: An Open Multilingual Graph of General Knowledge. In *AAAI*, 4444–4451.

Talmor, A.; Herzig, J.; Lourie, N.; and Berant, J. 2018. Commonsenseqa: A question answering challenge targeting commonsense knowledge. *arXiv preprint arXiv:1811.00937* .

Veličković, P.; Cucurull, G.; Casanova, A.; Romero, A.; LiÃš, P.; and Bengio, Y. 2018. Graph Attention Networks. In *International Conference on Learning Representations*. URL https://openreview.net/forum?id=rJXMpikCZ.

Wang, Z.; Bapst, V.; Heess, N.; Mnih, V.; Munos, R.; Kavukcuoglu, K.; and de Freitas, N. 2016. Sample efficient actor-critic with experience replay. *arXiv preprint arXiv:1611.01224* .

Winograd, T. 1972. Understanding natural language. *Cognitive psychology* 3(1): 1–191.

Yu, A. W.; Dohan, D.; Luong, M.-T.; Zhao, R.; Chen, K.; Norouzi, M.; and Le, Q. V. 2018. Qanet: Combining local convolution with global self-attention for reading comprehension. *arXiv preprint arXiv:1804.09541* .

Zahavy, T.; Haroush, M.; Merlis, N.; Mankowitz, D. J.; and Mannor, S. 2018. Learn what not to learn: Action elimination with deep reinforcement learning. In *Advances in Neural Information Processing Systems*, 3562–3573.