Interactive Fiction Game Playing as Multi-Paragraph Reading Comprehension with Reinforcement Learning

Xiaoxiao Guo[*]
IBM Research
xiaoxiao.guo@ibm.com

Mo Yu[*]
IBM Research
yum@us.ibm.com

Yupeng Gao
IBM Research
yupeng.gao@ibm.com

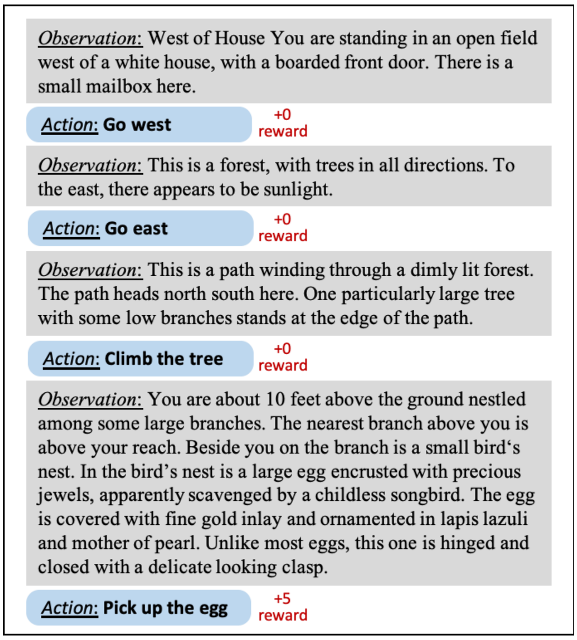
Chuang Gan
MIT-IBM Watson AI Lab
chuangg@ibm.com

Murray Campbell
IBM Research
mcam@us.ibm.com

Shiyu Chang
MIT-IBM Watson AI Lab
shiyu.chang@ibm.com

Abstract

Interactive Fiction (IF) games with real human-written natural language texts provide a new natural evaluation for language understanding techniques. In contrast to previous text games with mostly synthetic texts, IF games pose language understanding challenges on the human-written textual descriptions of diverse and sophisticated game worlds and language generation challenges on the action command generation from less restricted combinatorial space. We take a novel perspective of IF game solving and re-formulate it as Multi-Passage Reading Comprehension (MPRC) tasks. Our approaches utilize the context-query attention mechanisms and the structured prediction in MPRC to efficiently generate and evaluate action outputs and apply an object-centric historical observation retrieval strategy to mitigate the partial observability of the textual observations. Extensive experiments on the recent IF benchmark (*Jericho*) demonstrate clear advantages of our approaches achieving high winning rates and low data requirements compared to all previous approaches.[1]



Observation: West of House You are standing in an open field west of a white house, with a boarded front door. There is a small mailbox here.

Action: **Go west** +0 reward

Observation: This is a forest, with trees in all directions. To the east, there appears to be sunlight.

Action: **Go east** +0 reward

Observation: This is a path winding through a dimly lit forest. The path heads north south here. One particularly large tree with some low branches stands at the edge of the path.

Action: **Climb the tree** +0 reward

Observation: You are about 10 feet above the ground nestled among some large branches. The nearest branch above you is above your reach. Beside you on the branch is a small bird's nest. In the bird's nest is a large egg encrusted with precious jewels, apparently scavenged by a childless songbird. The egg is covered with fine gold inlay and ornamented in lapis lazuli and mother of pearl. Unlike most eggs, this one is hinged and closed with a delicate looking clasp.

Action: **Pick up the egg** +5 reward

Figure 1: Sample gameplay for the classic dungeon game *Zork1*. The objective is to solve various puzzles and collect the 19 treasures to install the trophy case. The player receives textual observations describing the current game state and additional reward scalars encoding the game designers' objective of game progress. The player sends textual action commands to control the protagonist.

1 Introduction

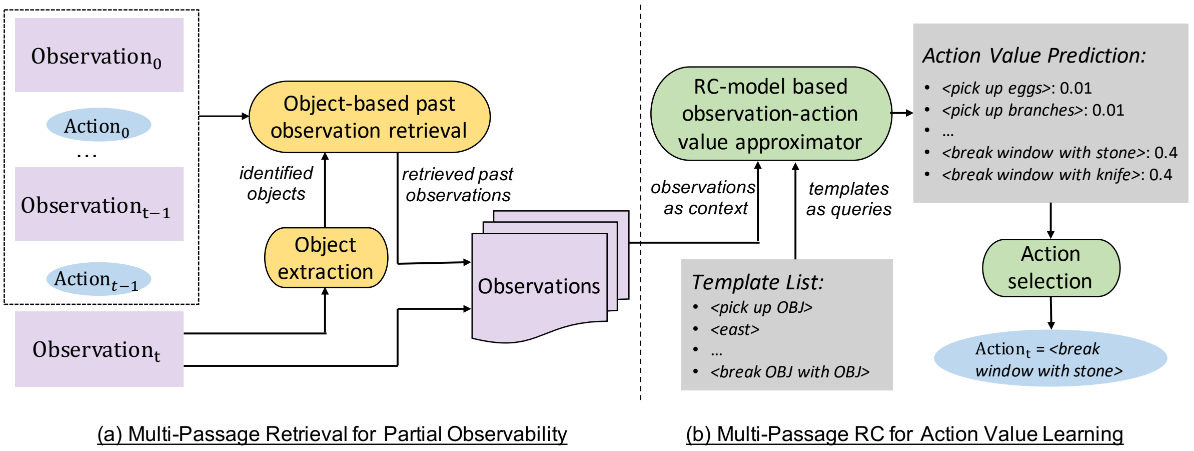
Interactive systems capable of understanding natural language and responding in the form of natural language text have high potentials in various applications. In pursuit of building and evaluating such systems, we study learning agents for Interactive Fiction (IF) games. IF games are world-simulating software in which players use text commands to control the protagonist and influence the world, as illustrated in Figure 1. IF gameplay agents need to simultaneously understand the game's information from a text display (**observation**) and generate natural language command (**action**) via a text input interface. Without providing an explicit game strategy, the agents need to identify behaviors that maximize objective-encoded cumulative rewards.

IF games composed of human-written texts (distinct from previous text games with synthetic texts) create superb new opportunities for studying and evaluating natural language understanding (NLU) techniques due to their unique characteristics. (1) Game designers elaborately craft on the literariness of the narrative texts to attract players when creating IF games. The resulted texts in IF games are more linguistically diverse and sophisticated than the template-generated ones in synthetic text games. (2) The language contexts of IF games

[*] Primary authors.

[1] Source code is available at: https://github.com/XiaoxiaoGuo/rcdqn.

Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, pages 7755–7765,
November 16–20, 2020. ©2020 Association for Computational Linguistics



(a) Multi-Passage Retrieval for Partial Observability

(b) Multi-Passage RC for Action Value Learning

Figure 2: Overview of our approach to solving the IF games as Multi-Paragraph Reading Comprehension (MPRC) tasks.

are more versatile because various designers contribute to enormous domains and genres, such as adventure, fantasy, horror, and sci-fi. (3) The text commands to control characters are less restricted, having sizes over six orders of magnitude larger than previous text games. The recently introduced *Jericho* benchmark provides a collection of such IF games (Hausknecht et al., 2019a).

The complexity of IF games demands more sophisticated NLU techniques than those used in synthetic text games. Moreover, the task of designing IF game-play agents, intersecting NLU and reinforcement learning (RL), poses several unique challenges on the NLU techniques. The first challenge is the difficulty of exploration in ***the huge natural language action space***. To make RL agents learn efficiently without prohibitive exhaustive trials, the action estimation must generalize learned knowledge from tried actions to others. To this end, previous approaches, starting with a single embedding vector of the observation, either predict the elements of actions independently (Narasimhan et al., 2015; Hausknecht et al., 2019a); or embed each valid action as another vector and predict action value based on the vector-space similarities (He et al., 2016). These methods do not consider the compositionality or role-differences of the action elements, or the interactions among them and the observation. Therefore, their modeling of the action values is less accurate and less data-efficient.

The second challenge is ***partial observability***. At each game-playing step, the agent receives a textual observation describing the locations, objects, and characters of the game world. But the latest observation is often not a sufficient summary of the interaction history and may not provide enough information to determine the long-term effects of actions. Previous approaches address this problem by building a representation over past observations (e.g., building a graph of objects, positions, and spatial relations) (Ammanabrolu and Riedl, 2019; Ammanabrolu and Hausknecht, 2020). These methods treat the historical observations equally and summarize the information into a single vector without focusing on important contexts related to the action prediction for the current observation. Therefore, their usages of history also bring noise, and the improvement is not always significant.

We propose a novel formulation of IF game playing as Multi-Passage Reading Comprehension (MPRC) and harness MPRC techniques to solve the *huge action space* and *partial observability* challenges. The graphical illustration is shown in Figure 2. First, the action value prediction (i.e., predicting the long-term rewards of selecting an action) is essentially *generating and scoring a compositional action structure by finding supporting evidence from the observation*. We base on the fact that each action is an instantiation of a **template**, i.e., a verb phrase with a few placeholders of object arguments it takes (Figure 2b). Then the action generation process can be viewed as extracting objects for a template's placeholders from the textual observation, based on the interaction between the template verb phrase and the relevant context of the objects in the observation. Our approach addresses the structured prediction and interaction problems with the idea of context-question attention mechanism in RC models. Specifically, we treat the observation as a passage and each template verb phrase as a question. The filling of object placeholders in the template thus becomes an extractive QA problem that selects objects from the observation given the template. Simultaneously each action (i.e., a template with all placeholder replaced) gets its evaluation value predicted by the RC model. Our formulation and approach better capture the fine-grained interactions between observation texts and structural actions, in contrast to previous approaches that represent the observation as a single vector and ignore the fine-grained dependency among action elements.

Second, alleviating partial observability is essentially *enhancing the current observation with potentially relevant history* and *predicting actions over the enhanced observation*. Our approach retrieves potentially relevant historical observations with an object-centric approach (Figure 2a), so that the retrieved ones are more likely to be connected to the current observation as they describe at least one shared interactable object. Our attention mechanisms are then applied across the retrieved multiple observation texts to focus on informative contexts for action value prediction.

We evaluated our approach on the suite of *Jericho* IF games, compared to all previous approaches. Our approaches achieved or outperformed the state-of-the-art performance on **25** out of 33 games, trained with less than **one-tenth** of game interaction data used by prior art. We also provided ablation studies on our models and retrieval strategies.

2 Related Work

IF Game Agents. Previous work mainly studies the text understanding and generation in parser-based or rule-based text game tasks, such as TextWorld platform (Côté et al., 2018) or custom domains (Narasimhan et al., 2015; He et al., 2016; Adhikari et al., 2020). The recent platform *Jericho* (Hausknecht et al., 2019a) supports over thirty human-written IF games. Earlier successes in real IF games mainly rely on heuristics without learning. NAIL (Hausknecht et al., 2019b) is the state-of-the-art among these "no-learning" agents, employing a series of reliable heuristics for exploring the game, interacting with objects, and building an internal representation of the game world. With the development of learning environments like *Jericho*, the RL-based agents have started to achieve dominating performance.

A critical challenge for learning-based agents is how to handle the **combinatorial action space** in IF games. LSTM-DQN (Narasimhan et al., 2015) was proposed to generate verb-object action with pre-defined sets of possible verbs and objects, but treat the selection and learning of verbs and objects independently. Template-DQN (Hausknecht et al., 2019a) extended LSTM-DQN for template-based action generation, introducing one additional but still independent prediction output for the second object in the template. Deep Reinforcement Relevance Network (DRRN) (He et al., 2016) was introduced for choice-based games. Given a set of valid actions at every game state, DRRN projects each action into a hidden space that matches the current state representation vector for action selection. Action-Elimination Deep Q-Network (AE-DQN) (Zahavy et al., 2018) learns to predict invalid actions in the adventure game *Zork*. It eliminates invalid action for efficient policy learning via utilizing expert demonstration data.

Other techniques focus on addressing the **partial observability** in text games. Knowledge Graph DQN (KG-DQN) (Ammanabrolu and Riedl, 2019) was proposed to deal with synthetic games. The method constructs and represents the game states as knowledge graphs with objects as nodes and uses pre-trained general purposed OpenIE tool and human-written rules to extract relations between objects. KG-DQN handles the action representation following DRRN. KG-A2C (Ammanabrolu and Hausknecht, 2020) later extends the work for IF games, by adding information extraction heuristics to fit the complexity of the object relations in IF games and utilizing a GRU-based action generator to handle the action space.

Reading Comprehension Models for Question Answering. Given a question, reading comprehension (RC) aims to find the answer to the question based on a paragraph that may contain supporting evidence. One of the standard RC settings is extractive QA (Rajpurkar et al., 2016; Joshi et al., 2017; Kwiatkowski et al., 2019), which extracts a span from the paragraph as an answer. Our formulation of IF game playing resembles this setting.

Many neural *reader* models have been designed for RC. Specifically, for the extractive QA task, the reader models usually build question-aware passage representations via attention mechanisms (Seo et al., 2016; Yu et al., 2018), and employ a pointer network to predict the start and end positions of the answer span (Wang and Jiang, 2016). Powerful pre-trained language models (Peters et al., 2018; Devlin et al., 2019; Radford et al., 2019) have been recently applied to enhance the encoding and attention mechanisms of the aforementioned reader models. They give performance boost but are more resource-demanding and do not suit the IF game playing task very well.

Reading Comprehension over Multiple Paragraphs. Multi-paragraph reading comprehension (MPRC) deals with the more general task of answering a question from multiple related paragraphs, where each paragraph may not necessarily support the correct answer. Our formulation becomes an MPRC setting when we enhance the state representation with historical observations and predict actions from multiple observation paragraphs.

A fundamental research problem in MPRC, which is also critical to our formulation, is to select relevant paragraphs from all the input paragraphs for the reader to focus on. Previous approaches mainly apply traditional IR approaches like BM25 (Chen et al., 2017; Joshi et al., 2017), or neural ranking models trained with distant supervision (Wang et al., 2018; Min et al., 2019a), for paragraph selection. Our formulation also relates to the work of evidence aggregation in MPRC (Wang et al., 2017; Lin et al., 2018), which aims to infer the answers based on the joint of evidence pieces from multiple paragraphs. Finally, recently some works propose the entity-centric paragraph retrieval approaches (Ding et al., 2019; Godbole et al., 2019; Min et al., 2019b; Asai et al., 2019), where paragraphs are connected if they share the same-named entities. The paragraph retrieval then becomes a traversal over such graphs via entity links. These entity-centric paragraph retrieval approaches share a similar high-level idea to our object-based history retrieval approach. The techniques above have been applied to deal with evidence from Wikipedia, news collections, and, recently, books (Mou et al., 2020). We are the first to extend these ideas to IF games.

3 Multi-Paragraph RC for IF Games

3.1 Problem Formulation

Each IF game can be defined as a Partially Observable Markov Decision Process (POMDP), namely a 7-tuple of $\langle\ S,\ A,\ T,\ O,\ \Omega,\ R,\ \gamma\ \rangle$, representing the hidden game state set, the action set, the state transition function, the set of textual observations composed from vocabulary words, the textual observation function, the reward function, and the discount factor respectively. The game playing agent interacts with the game engine in multiple turns until the game is over or the maximum number of steps is reached. At the t-th turn, the agent receives a textual observation describing the current game state $o_t \in O$ and sends a textual action command $a_t \in A$ back. The agent receives additional reward scalar r_t which encodes the game designers' objective of game progress. Thus the task of the game playing can be formulated to generate a textual action command per step as to maximize the expected cumulative discounted rewards $\mathbf{E}\left[\sum_{t=0}^{\infty}\gamma^t r_t\right]$. Value-based RL approaches learn to approximate an observation-action value function $Q(o_t, a_t; \boldsymbol{\theta})$ which measures the expected cumulative rewards of taking action a_t when observing o_t. The agent selects action based on the action value prediction of $Q(o, a; \boldsymbol{\theta})$.

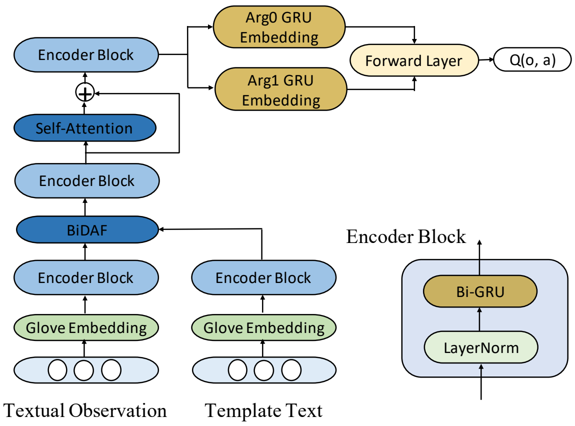


Figure 3: Our RC-based action prediction model architecture. The template text is a verb phrase with placeholders for objects, such as [pick up OBJ] and [break OBJ with OBJ].

Template Action Space. Template action space considers actions satisfying decomposition in the form of $\langle verb, arg_0, arg_1\rangle$. *verb* is an interchangeable verb phrase template with placeholders for objects and arg_0 and arg_1 are optional objects. For example, the action command [east], [pick up eggs] and [break window with stone] can be represented as template actions $\langle east, none, none\rangle$, $\langle pick\ up\ OBJ, eggs, none$ and $\langle break\ OBJ\ with\ OBJ, window, stone\rangle$. We reuse the template library and object list from *Jericho*. The verb phrases usually consist of several vocabulary words and each object is usually a single word.

3.2 RC Model for Template Actions

We parameterize the observation-action value function $Q(o, a{=}\langle \mathit{verb}, \mathit{arg}_0, \mathit{arg}_1\rangle; \boldsymbol{\theta})$ by utilizing the decomposition of the template actions and context-query contextualized representation in RC. Our model treats the observation o as a context in RC and the $\mathit{verb}{=}(v_1, v_2, ..., v_k)$ component of the template actions as a query. Then a *verb*-aware observation representation is derived via a RC reader model with Bidirectional Attention Flow (BiDAF) (Seo et al., 2016) and self-attention. The observation representation responding to the arg_0 and arg_1 words are pooled and projected to a scalar value estimate for $Q(o, a{=}\langle \mathit{verb}, \mathit{arg}_0, \mathit{arg}_1\rangle; \boldsymbol{\theta})$. A high-level model architecture of our model is illustrated in Figure 3.

Observation and *verb* Representation. We tokenize the observation and the *verb* phrase into words, then embed these words using pre-trained GloVe embeddings (Pennington et al., 2014). A shared encoder block that consists of LayerNorm (Ba et al., 2016) and Bidirectional GRU (Cho et al., 2014) processes the observation and *verb* word embeddings to obtain the separate observation and *verb* representation.

Observation-*verb* Interaction Layers. Given the separate observation and *verb* representation, we apply two attention mechanisms to compute a *verb*-contextualized observation representation. We first apply BiDAF with observation as the context input and *verb* as the query input. Specifically, we denote the processed embeddings for observation word i and template word j as $\boldsymbol{o_i}$ and $\boldsymbol{t}_j$. The attention between the two words is then $a_{ij}{=}\boldsymbol{w_1}{\cdot}\boldsymbol{o_i}{+}\boldsymbol{w_2}{\cdot}\boldsymbol{t_j}{+}\boldsymbol{w_3}{\cdot}(\boldsymbol{o_i}{\otimes}\boldsymbol{t_j})$, where $\boldsymbol{w_1}$, $\boldsymbol{w_2}$, $\boldsymbol{w_3}$ are learnable vectors and $\otimes$ is element-wise product. We then compute the "*verb*2observation" attention vector for the i-th observation word as $\boldsymbol{c_i}{=}\sum_j p_{ij}\boldsymbol{t_j}$ with $p_{ij}{=}\exp(a_{ij})/\sum_j \exp(a_{ij})$. Similarly, we compute the "observation2*verb*" attention vector as $\boldsymbol{q}{=}\sum_i p_i\boldsymbol{o_i}$ with $p_i = \exp(\max_j a_{ij})/\sum_i \exp(\max_j a_{ij})$. We concatenate and project the output vectors as $\boldsymbol{w_4} \cdot [\boldsymbol{o_i}, \boldsymbol{c_i}, \boldsymbol{o_i} \otimes \boldsymbol{c_i}, \boldsymbol{q} \otimes \boldsymbol{c_i}]$, followed by a linear layer with leaky ReLU activation units (Maas et al., 2013). The output vectors are processed by an encoder block. We then apply a residual self-attention on the outputs of the encoder block. The self-attention is the same as BiDAF, but only between the observation and itself.

Observation-Action Value Prediction. We generate an action by replacing the placeholders (arg_0 and arg_1) in a template with objects appearing in the observation. The observation-action value $Q(o, a{=}\langle \mathit{verb}, \mathit{arg}_0{=}\mathit{obj}_m, \mathit{arg}_1{=}\mathit{obj}_n\rangle; \theta)$ is achieved by processing each object's corresponding *verb*-contextualized observation representation. Specifically, we get the indices of an *obj* in the observation texts $I(\mathit{obj}, o)$. When the object is a noun phrase, we take the index of its headword.[2] Because the same object has different meanings when it replaces different placeholders, we apply two GRU-based embedding functions for the two placeholders, to get the object's *verb*-placeholder dependent embeddings. We derive a single vector representation $\boldsymbol{h}_{\mathit{arg}_0=\mathit{obj}_m}$ for the case that the placeholder arg_0 is replaced by obj_m by mean-pooling over the *verb*-placeholder dependent embeddings indexed by $I(\mathit{obj}_m, o)$ for the corresponding placeholder arg_0. We apply a linear transformation on the concatenated embeddings of the two placeholders to obtain the observation action value $Q(o, a){=}\boldsymbol{w}_5 \cdot [\boldsymbol{h}_{\mathit{arg}_0=\mathit{obj}_m}, \boldsymbol{h}_{\mathit{arg}_1=\mathit{obj}_n}]$ for $a{=}\langle \mathit{verb}, \mathit{arg}_0{=}\mathit{obj}_m, \mathit{arg}_1{=}\mathit{obj}_n\rangle$. Our formulation avoids the repeated computation overhead among different actions with a shared template verb phrase.

3.3 Multi-Paragraph Retrieval Method for Partial Observability

The observation at the current step sometimes does not have full-textual evidence to support action selection and value estimation, due to the inherent partial observability of IF games. For example, when repeatedly attacking a troll with a sword, the player needs to know the effect or feedback of the last attack to determine if an extra attack is necessary. It is thus important for an agent to efficiently utilize historical observations to better support action value prediction. In our RC-based action prediction model, the historical observation utilization can be formulated as selecting evidential observation paragraphs in history, and predicting the action values from multiple selected observations, namely a Multiple-Paragraph Reading Comprehension (MPRC) problem. We propose to retrieve past observations with an object-centric approach.

Past Observation Retrieval. Multiple past observations may share objects with the current obser-

[2] Some templates may take zero or one object. We denote the unrequired objects as `none` so that all templates take two objects. The index of the *none* object is for a special token. We set to the index of split token of the observation contents.

Agents	Action strategy	State strategy	Interaction data
TDQN	Independent selection of template and the two objects	*None*	1M
DRRN	Action as a word sequence without distinguishing the roles of verbs and objects	*None*	1M
KG-A2C	Recurrent neural decoder that selects the template and objects in a fixed order	Object graph from historical observations based on OpenIE and human-written rules	1.6M
Ours	Observation-template representation for object-centric value prediction	Object-based history observation retrieval	0.1M

Table 1: Summary of the main technical differences between our agent and the baselines. All agents use DQN to update the model parameters except KG-A2C uses A2C. All agents use the same handicaps.

vation, and it is computationally expensive and unnecessary to retrieve all of such observations. The utility of past observations associated with each object is often time-sensitive in that new observations may entirely or partially invalidate old observations. We thus propose a time-sensitive strategy for retrieving past observations. Specifically, given the detected objects from the current observation, we retrieve the most recent K observations with at least one shared object. The K retrieved observations are sorted by time steps and concatenated to the current observation. The observations from different time steps are separated by a special token. Our RC-based action prediction model treats the concatenated observations as the observation inputs, and no other parts are changed. We use the notation o_t to represent the current observation and the extended current observation interchangeably.

3.4 Training Loss

We apply the Deep Q-Network (DQN) (Mnih et al., 2015) to update the parameters $\boldsymbol{\theta}$ of our RC-based action prediction model. The loss function is:

$$\begin{aligned}\mathcal{L}(\theta) = \mathbf{E}_{(o_t,a_t,r_t,o_{t+1})\sim\rho(\mathcal{D})}\Big[||Q(o_t, a_t; \theta) \\ - (r_t + \gamma \max_b Q(o_{t+1}, b; \theta^-))||\Big]\end{aligned}$$

where $\mathcal{D}$ is the experience replay consisting of recent gameplay transition records and ρ is a distribution over the transitions defined by a sampling strategy.

Prioritized Trajectories. The distribution ρ has a decent impact on DQN performance. Previous work samples transition tuples with immediate positive rewards more frequently to speed up learning (Narasimhan et al., 2015; Hausknecht et al., 2019a). We observe that this heuristic is often insufficient. Some transitions with zero immediate rewards or even negative rewards are also indispensable in recovering well-performed trajectories. We thus extend the strategy from transition level to trajectory level. We prioritize transitions from trajectories that outperform the exponential moving average score of recent trajectories.

4 Experiments

We evaluate our proposed methods on the suite of Jericho supported games. We compared to all previous baselines that include recent methods addressing the huge action space and partial observability challenges.

4.1 Setup

***Jericho* Handicaps and Configuration.** The handicaps used by our methods are the same as other baselines. First, we use the Jericho API to check if an action is valid with game-specific templates. Second, we augmented the observation with the textual feedback returned by the command [*inventory*] and [*look*]. Previous work also included the last action or game score as additional inputs. Our model discarded these two types of inputs as we did not observe a significant difference by our model. The maximum game step number is set to 100 following baselines.

Implementation Details. We apply spaCy[3] to tokenize the observations and detect the objects in the observations. We use the 100-dimensional GloVe embeddings as fixed word embeddings. The out-of-vocabulary words are mapped to a randomly initialized embedding. The dimension of Bi-GRU hidden states is 128. We set the observation representation dimension to be 128 throughout the model. The history retrieval window K is 2. For DQN configuration, we use the ϵ-greedy strategy

[3] `https://spacy.io`

		Human		Baselines		Ours	
Game	**Max**	**Walkthrough-100**	**TDQN**	**DRRN**	**KG-A2C**	**MPRC-DQN**	**RC-DQN**
905	1	1	**0**	**0**	**0**	**0**	**0**
acorncourt	30	30	1.6	**10**	0.3	**10.0**	**10.0**
advent	350	113	36	36	36	**63.9**	36
adventureland	100	42	0	20.6	0	**24.2**	21.7
afflicted	75	75	1.3	2.6	–	**8.0**	**8.0**
anchor	100	11	**0**	**0**	**0**	**0**	**0**
awaken	50	50	**0**	**0**	**0**	**0**	**0**
balances	51	30	4.8	**10**	**10**	**10**	**10**
deephome	300	83	**1**	**1**	**1**	**1**	**1**
detective	360	350	169	197.8	207.9	**317.7**	291.3
dragon	25	25	-5.3	-3.5	0	0.04	**4.84**
enchanter	400	125	8.6	**20**	12.1	**20.0**	**20.0**
gold	100	30	**4.1**	0	–	0	0
inhumane	90	70	0.7	0	**3**	0	0
jewel	90	24	0	1.6	1.8	**4.46**	2.0
karn	170	40	0.7	2.1	0	**10.0**	**10.0**
library	30	30	6.3	17	14.3	17.7	**18.1**
ludicorp	150	37	6	13.8	17.8	**19.7**	17.0
moonlit	1	1	**0**	**0**	**0**	**0**	**0**
omniquest	50	50	**16.8**	10	3	10.0	10.0
pentari	70	60	17.4	27.2	**50.7**	44.4	43.8
reverb	50	50	0.3	**8.2**	–	2.0	2.0
snacktime	50	50	**9.7**	0	0	0	0
sorcerer	400	150	5	20.8	5.8	**38.6**	38.3
spellbrkr	600	160	18.7	**37.8**	21.3	25	25
spirit	250	8	0.6	0.8	1.3	3.8	**5.2**
temple	35	20	7.9	7.4	7.6	**8.0**	**8.0**
tryst205	350	50	0	9.6	–	**10.0**	**10.0**
yomomma	35	34	0	0.4	–	**1.0**	**1.0**
zenon	20	20	0	0	**3.9**	0	0
zork1	350	102	9.9	32.6	34	38.3	**38.8**
zork3	7	3[a]	0	0.5	0.1	**3.63**	2.83
ztuu	100[b]	100	4.9	21.6	9.2	**85.4**	79.1
Winning percentage / counts			24%/8	30%/10	27%/9	**64%/21**	52%/17
						76%/25	

Table 2: Average game scores on Jericho benchmark games. The best performing agent score per game is **in bold**. The *Winning percentage / counts* row computes the percentage / counts of games that the corresponding agent is best. The scores of baselines are from their papers. The missing scores are represented as "–", for which games KG-A2C skipped. We also added the 100-step results from a human-written game-playing walkthrough, as a reference of human-level scores. We denote the difficulty levels of the games defined in the original Jericho paper with colors in their names – possible (i.e., easy or normal) games in green color, difficult games in tan and extreme games in red. Best seen in color.

[a] *Zork3* walkthrough does not maximize the score in the first 100 steps but explores more. [b] Our agent discovers some unbounded reward loops in the game *Ztuu*.

for exploration, annealing ϵ from 1.0 to 0.05. γ is 0.98. We use Adam to update the weights with 10^{-4} learning rate. Other parameters are set to their default values. More details of the Reproducibility Checklist is in Appendix A.

Baselines. We compare with all the public results on the Jericho suite, namely TDQN (Hausknecht et al., 2019a), DRRN (He et al., 2016), and KG-A2C (Ammanabrolu and Hausknecht, 2020). As discussed, our approaches differ from them mainly in the strategies of handling the large action space and partial observability of IF games. We summarize these main technical differences in Table 1. In summary, all previous agents predict actions conditioned on a single vector representation of the whole observation texts. Thus they do not exploit the fine-grained interplay among the template components and the observations. Our approach addresses this problem by formulating action prediction as an RC task, better utilizing the rich textual observations with deeper language understanding.

Training Sample Efficiency. We update our models for 100,000 times. Our agents interact with the environment one step per update, resulting in a total of 0.1M environment interaction data. Compared to the other agents, such as KG-A2C (1.6M), TDQN (1M), and DRRN (1M), our environment interaction data is significantly smaller.

Game	Template Action Space ($\times 10^6$)	Avg. Steps Per Reward	Dialog Actions	Darkness Limit	Nonstandard	Inventory
advent	107	7		✓	✓	✓
detective	19	2				
karn	63	17	✓	✓		
ludicorp	45	4			✓	✓
pentari	32	5			✓	
spirit	195	21	✓	✓	✓	✓
zork3	67	39	✓	✓		✓

Table 3: Difficulty levels and characteristics of games on which our approach achieves the most considerable improvement. *Dialog* indicates that it is necessary to speak with another character. *Darkness* indicates that accessing some dark areas requires a light source. *Nonstandard Actions* refers to actions with words not in an English dictionary. *Inventory Limit* restricts the number of items carried by the player. Please refer to (Hausknecht et al., 2019a) for more comprehensive definitions.

4.2 Overall Performance

We summarize the performance of our Multi-Paragraph Reading Comprehension DQN (MPRC-DQN) agent and baselines in Table 2. Of the 33 IF games, our MPRC-DQN achieved or improved the state of the art performance on 21 games (i.e., a winning rate of 64%). The best performing baseline (DRRN) achieved the state-of-the-art performance on only ten games, corresponding to the winning rate of 30%, lower than half of ours. Note that all the methods achieved the same initial scores on five games, namely *905*, *anchor*, *awaken*, *deephome*, and *moonlit*. Apart from these five games, our MPRC-DQN achieved more than three times wins. Our MPRC-DQN achieved significant improvement on some games, such as *adventureland*, *afflicted*, *detective*, etc. Appendix C shows some game playing trajectories.

We include the performance of an RC-DQN agent, which implements our RC-based action prediction model but only takes the current observations as inputs. It also outperformed the baselines by a large margin. After we consider the RC-DQN agent, our MPRC-DQN still has the highest winning percentage, indicating that our RC-based action prediction model has a significant impact on the performance improvement of our MPRC-DQN and the improvement from the multi-passage retrieval is also unneglectable. Moreover, compared to RC-DQN, our MPRC-DQN has another advantage of faster convergence. The learning curves of our MPRC-DQN and RC-DQN agents on various games are in Appendix B.

Finally, our approaches, overall, achieve the new state-of-the-art on 25 games (i.e., a winning rate of 76%), giving a significant advance in the field of IF game playing.

Competitors	Win	Draw	Lose
MPRC-DQN v.s. TDQN	23	6	4
MPRC-DQN v.s. DRRN	18	13	2
MPRC-DQN v.s. KG-A2C	18	7	3

Table 4: Pairwise comparison between our MPRC-DQN versus each baseline.

Pairwise Competition. To better understand the performance difference between our approach and each of the baselines, we adopt a direct one-to-one comparison metric based on the results from Table 2. Our approach has a high winning rate when competing with any of the baselines, summarized in Table 4. All the baselines have a rare chance to beat us on games. DRRN gives a higher chance of draw-games when competing with ours.

Human-Machine Gap. We additionally compare IF gameplay agents to human players to better understand the improvement significance and the potential improvement upper-bound. We measure each agent's game progress as the macro-average of the normalized agent-to-human game score ratios, capped at 100%. The progress of our MPRC-DQN is 28.5%, while the best performing baseline DRRN is 17.8%, showing that our agent's improvement is significant even in the realm of human players. Nevertheless, there is a vast gap between the learning agents and human players. The gap indicates IF games can be a good benchmark for the development of natural language understanding techniques.

Difficulty Levels of Games. Jericho categorizes the supported games into three difficulty levels, namely possible games, difficult games, and extreme games, based on the characteristics of the game dynamics, such as the action space size, the length of the game, and the average number of

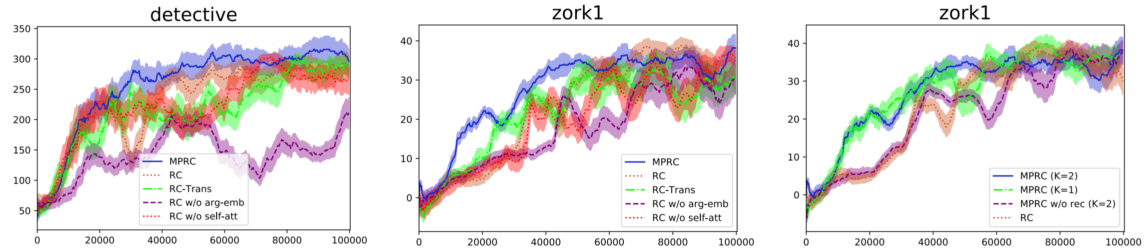


Figure 4: Learning curves for ablative studies. **(left)** Model ablative studies on the game *Detective*. **(middle)** Model ablative studies on *Zork1*. **(right)** Retrieval strategy study on *Zork1*. Best seen in color.

steps to receive a non-zero reward. Our approach improves over prior art on seven of the sixteen possible games, seven of the eleven difficult games, and three of the six extreme games in Table 2. It shows that the strategies of our method are generally beneficial for any difficulty levels of game dynamics. Table 3 summarizes the characteristics of the seven games in which our method improves the most, i.e., larger than 15% of the game progress in the first 100 steps.[4] First, these mostly improved games have medium action space sizes, and it is an advantageous setting for our methods where modeling the template-object-observation interactions is effective. Second, our approach improves most on games with a reasonably high degree of reward sparsity, such as *karn*, *spirit*, and *zork3*, indicating that our RC-based value function formulation helps in optimization and mitigates the reward sparsity. Finally, we remark that these game difficulty levels are not directly categorized based on natural language-related characteristics, such as text comprehension and puzzle-solving difficulties. Future studies on additional game categories based on those natural language-related characteristics would shed light on related improvements.

4.3 Ablative Studies

RC-model Design. The overall results show that our RC-model plays a critical role in performance improvement. We compare our RC-model to some alternative models as ablative studies. We consider three alternatives, namely (1) our RC-model without the self-attention component (`w/o self-att`), (2) without the argument-specific embedding (`w/o arg-emb`) and (3) our RC-model with Transformer-based block encoder (`RC-Trans`) following QANet (Yu et al., 2018). Detailed architecture is in Appendix A.

The learning curves for different RC-models are in Figure 4 (left/middle). The RC-models without either self-attention or argument-specific embedding degenerate, and the argument-specific embedding has a greater impact. The Transformer-based encoder block sometimes learns faster than Bi-GRU at the early learning stage. It achieved a comparable final performance, even with much greater computational resource requirements.

Retrieval Strategy. We compare with history retrieval strategies with different history sizes (K) and pure recency-based strategies (i.e., taking the latest K observations as history, denoted as `w/o rec`). The learning curves of different strategies are in Figure 4 (right). In general, the impact of history window size is highly game-dependent, but the pure recency based ones do not differ significantly from RC-DQN at the beginning of learning. The issues of pure recency based strategy are: (1) limited additional information about objects provided by successive observations; and (2) higher variance of retrieved observations due to policy changes.

5 Conclusion

We formulate the general IF game playing as MPRC tasks, enabling an MPRC-style solution to efficiently address the key IF game challenges on the huge combinatorial action space and the partial observability in a unified framework. Our approaches achieved significant improvement over the previous state-of-the-art on both game scores and training data efficiency. Our formulation also bridges broader NLU/RC techniques to address other critical challenges in IF games for future work, e.g., common-sense reasoning, novelty-driven exploration, and multi-hop inference.

Acknowledgments

We would like to thank Matthew Hausknecht for helpful discussions on the Jericho environments.

[4] We ignore *ztuu* due to the infinite reward loops.

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*.

Prithviraj Ammanabrolu and Matthew Hausknecht. 2020. Graph constrained reinforcement learning for natural language action spaces. *arXiv*, pages arXiv–2001.

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.

Akari Asai, Kazuma Hashimoto, Hannaneh Hajishirzi, Richard Socher, and Caiming Xiong. 2019. Learning to retrieve reasoning paths over wikipedia graph for question answering. *arXiv preprint arXiv:1911.10470*.

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. 2016. Layer normalization. *arXiv preprint arXiv:1607.06450*.

Danqi Chen, Adam Fisch, Jason Weston, and Antoine Bordes. 2017. Reading wikipedia to answer open-domain questions. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1870–1879.

Kyunghyun Cho, Bart Van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio. 2014. On the properties of neural machine translation: Encoder-decoder approaches. *arXiv preprint arXiv:1409.1259*.

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

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186.

Ming Ding, Chang Zhou, Qibin Chen, Hongxia Yang, and Jie Tang. 2019. Cognitive graph for multi-hop reading comprehension at scale. In *Proceedings of ACL 2019*.

Ameya Godbole, Dilip Kavarthapu, Rajarshi Das, Zhiyu Gong, Abhishek Singhal, Hamed Zamani, Mo Yu, Tian Gao, Xiaoxiao Guo, Manzil Zaheer, et al. 2019. Multi-step entity-centric information retrieval for multi-hop question answering. *arXiv preprint arXiv:1909.07598*.

Matthew Hausknecht, Prithviraj Ammanabrolu, Marc-Alexandre Côté, and Xingdi Yuan. 2019a. Interactive fiction games: A colossal adventure. *arXiv preprint arXiv:1909.05398*.

Matthew Hausknecht, Ricky Loynd, Greg Yang, Adith Swaminathan, and Jason D Williams. 2019b. Nail: A general interactive fiction agent. *arXiv preprint arXiv:1902.04259*.

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.

Mandar Joshi, Eunsol Choi, Daniel S Weld, and Luke Zettlemoyer. 2017. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1601–1611.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, et al. 2019. Natural questions: a benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:453–466.

Yankai Lin, Haozhe Ji, Zhiyuan Liu, and Maosong Sun. 2018. Denoising distantly supervised open-domain question answering. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1736–1745.

Andrew L Maas, Awni Y Hannun, and Andrew Y Ng. 2013. Rectifier nonlinearities improve neural network acoustic models. In *Proc. icml*, volume 30, page 3.

Sewon Min, Danqi Chen, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2019a. A discrete hard em approach for weakly supervised question answering. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2844–2857.

Sewon Min, Danqi Chen, Luke Zettlemoyer, and Hannaneh Hajishirzi. 2019b. Knowledge guided text retrieval and reading for open domain question answering. *arXiv preprint arXiv:1911.03868*.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. 2015. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533.

Xiangyang Mou, Mo Yu, Bingsheng Yao, Chenghao Yang, Xiaoxiao Guo, Saloni Potdar, and Hui Su. 2020. Frustratingly hard evidence retrieval for qa over books. In *Proceedings of the First Joint Workshop on Narrative Understanding, Storylines, and Events*, pages 108–113.

Karthik Narasimhan, Tejas Kulkarni, and Regina Barzilay. 2015. Language understanding for text-based games using deep reinforcement learning. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1–11.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

Matthew E Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of NAACL-HLT*, pages 2227–2237.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392.

Minjoon Seo, Aniruddha Kembhavi, Ali Farhadi, and Hannaneh Hajishirzi. 2016. Bidirectional attention flow for machine comprehension.

Shuohang Wang and Jing Jiang. 2016. Machine comprehension using match-lstm and answer pointer. *arXiv preprint arXiv:1608.07905*.

Shuohang Wang, Mo Yu, Xiaoxiao Guo, Zhiguo Wang, Tim Klinger, Wei Zhang, Shiyu Chang, Gerry Tesauro, Bowen Zhou, and Jing Jiang. 2018. R 3: Reinforced ranker-reader for open-domain question answering. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

Shuohang Wang, Mo Yu, Jing Jiang, Wei Zhang, Xiaoxiao Guo, Shiyu Chang, Zhiguo Wang, Tim Klinger, Gerald Tesauro, and Murray Campbell. 2017. Evidence aggregation for answer re-ranking in open-domain question answering. *arXiv preprint arXiv:1711.05116*.

Adams Wei Yu, David Dohan, Minh-Thang Luong, Rui Zhao, Kai Chen, Mohammad Norouzi, and Quoc V Le. 2018. Qanet: Combining local convolution with global self-attention for reading comprehension. *arXiv preprint arXiv:1804.09541*.

Tom Zahavy, Matan Haroush, Nadav Merlis, Daniel J Mankowitz, and Shie Mannor. 2018. Learn what not to learn: Action elimination with deep reinforcement learning. In *Advances in Neural Information Processing Systems*, pages 3562–3573.