PLM-based World Models for Text-based Games

Minsoo Kim[♠] **YeonJoon Jung**[♣] **Dohyeon Lee**[♢] **Seung-won Hwang**[*♢]

[♠]Interdisciplinary Program in Artificial Intelligence, Seoul National University

[♢]Department of Computer Science and Engineering, Seoul National University

[♣]Department of Artificial Intelligence, Yonsei University

{minsoo9574, waylight3, seungwonh}@snu.ac.kr

{theaitetus}@yonsei.ac.kr

Abstract

World models have improved the ability of reinforcement learning agents to operate in a sample efficient manner, by being trained to predict plausible changes in the underlying environment. As the core tasks of world models are future prediction and commonsense understanding, our claim is that pre-trained language models (PLMs) already provide a strong base upon which to build world models. Worldformer is a recently proposed world model for text-based game environments, based only partially on PLM and transformers. Our distinction is to fully leverage PLMs as actionable world models in text-based game environments, by reformulating generation as constrained decoding which decomposes actions into verb templates and objects. We show that our model improves future valid action prediction and graph change prediction.[1] Additionally, we show that our model better reflects commonsense than standard PLM.

1 Introduction

In model-based reinforcement learning, world models (Ha and Schmidhuber, 2018), being trained to predict plausible changes in the underlying environment, has helped agents to quickly identify sensible or high-value actions. As a result, agents equipped with world models have outperformed state-of-the art model-free algorithms in Atari games, while drastically improving sample efficiency (Kaiser et al., 2019; Hafner et al., 2020).

In this work, we focus on world modeling in text-based game (TBG) environments, in which players and agents must perceive and interact with the world entirely through textual natural language. As such, they present several unique challenges (Côté et al., 2018; Hausknecht et al., 2019; Ammanabrolu and Riedl, 2021b): Aside from language understanding itself, TBGs require dealing with an exponential action space: For instance, the action space size for Zork1, combining five words from a vocabulary of 697 words, is $\mathcal{O}(10^{14})$. TBGs also require rich and accurate knowledge representations of locations and objects, in order to facilitate navigation and interaction. Finally, solving TBGs requires commonsense reasoning, including understanding object affordance, and the causal ramifications of actions. The first dataset and model for learning world models in TBGs were proposed by JerichoWorld (Ammanabrolu and Riedl, 2021b), and Worldformer (Ammanabrolu and Riedl, 2021a), respectively. Intuitively, large pre-trained language models (PLMs), make promising candidates for instantiating world models (Yao et al., 2020), as they are already trained to generate language in a much larger action space, e.g. $\mathcal{O}(40,000^{512})$, assuming a standard vocabulary of $40,000$ tokens, and context length of 512 tokens. While promising, adapting PLMs to TBGs is non-trivial, as in

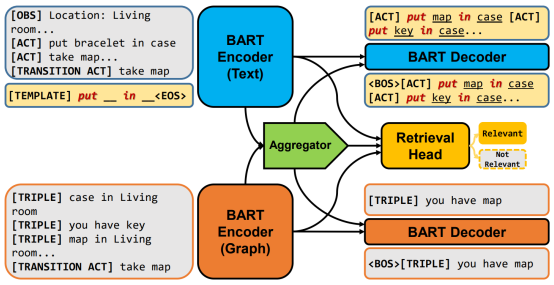

Figure 1: Model architecture of AWM-BART, illustrating the dual world modeling tasks of JerichoWorld. BART encoder-decoder denoted in blue learns the *future valid actions prediction* task, while the one denoted in orange simultaneously learns the *graph difference prediction* task. The light yellow blocks denote generation constrained by a template, bringing generation closer to the target *controlled sublanguage* of the text-based game's parser.

[*]Corresponding author.

[1]Code and data are available at: https://github.com/mnskim/awm-bart

Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, pages 1324–1341
December 7-11, 2022

TBGs, generated actions must be executable in the game environment. While we are not generating in a formal syntax as in semantic parsing, we are still constrained by a *controlled sublanguage* (Shin et al., 2021) , which is closer to natural language but follows a grammar defined by the engine's parser.

Furthermore, a world modeling task such as the valid action decoding task of JerichoWorld, requires predicting, from a particular world state, all *future valid actions*. In contrast, semantic parsing focuses on learning a one-to-one mapping between natural language and the controlled sublanguage. To encapsulate these challenges, we define a concept of **actionability** as the main objective of the valid action generation task of JerichoWorld, in that the model's generation should be actionable in the TBG environment, by 1) conforming to the *parseable* controlled sublanguage, and 2) ensuring the generation of *valid actions* consistent with the commonsense governing the dynamics of the TBG environment. In this work, we focus on improving the actionability of the world model's generations, focusing in particular on bringing the PLM's commonsense and reasoning capabilities to bear on world modeling tasks.

We begin by building Worldformer-BART, an implementation of Worldformer based on BART. While successful, commonsense in PLM-based world models are not easily transferred to more formal forms, such as that of the controlled sublanguage of TBGs. Motivated by these findings, we build a retrieval-augmented model which aims to enhance actionability by formalizing commonsense through templates, hence named **Actionable World Model (AWM)**. We show through experimental results that AWM-BART significantly improves actionability over Worldformer-BART. We also compare with augmenting trained models with a COMET-based commonsense filter, showing that our model outperforms such approaches.

2 Background

2.1 Text World Environments

Text world environments are typically modeled as Partially-Observable Markov Decision Processes (POMDPs) (Côté et al., 2018; Hausknecht et al., 2019), defined by the tuple $\langle S, A, T, \Omega, O, R, , \gamma \rangle$. Respectively, each item in the tuple corresponds to the set of environment states, the text-based action that changes the game state according to a transition function, the mostly-deterministic latent transition function, observation conditional probabilities, the observations, i.e. the game's text responses, the reward function, and the discount factor.

While state-of-the-art agents can be trained on these environments using model-free RL algorithms, they often rely on large amounts of interaction with the environment (Kaiser et al., 2019; Yarats et al., 2021). In contrast, model-based learning proposes to learn a predictive model of the environment, also known as a *world model*, to aid an agent to learn the underlying dynamics of the game, and better predict which actions will lead to desirable outcomes. These approaches are closely inspired by research on human cognitive processes, which hypothesize that human decision-making is directly influenced by an internal predictive model of the future (Ha and Schmidhuber, 2018). For world models, recurrent neural networks are a suitable solution to overcome the partial observability of the environment in POMDPs, and in text-based games environments, pre-trained language models are promising candidates.

Popular text-based game environments include TextWorld (Côté et al., 2018), which provides procedurally generated environments, allowing for the complexity and content of the generated game to be variable, LIGHT (Urbanek et al., 2019), a large-scale crowdsourced text adventure game, whose dataset provides agent-to-agent dialogs to study grounded social interactions, and Jericho (Hausknecht et al., 2019), a collection of 32 diverse human-made interactive fiction games, covering a wide range of genres.

2.2 JerichoWorld

Different from the aforementioned environments, JerichoWorld (Ammanabrolu and Riedl, 2021b) is the first dataset specifically targeting the learning of world models in text-based game environments. JerichoWorld is generated by simulating playthroughs of Jericho games, based on human-generated gold walkthroughs, combined with random exploration to increase the coverage of the state spaces of games. Each example in the dataset is a tuple of the form, $\langle S_t, A_t, S_{t+1}, R \rangle$ consisting of a previous state S_t, a transition action A_t, the next state S_{t+1}, and the observed reward R. A key feature of the dataset is that it maps text observations to both knowledge graphs, which consist of a set of $\langle s, r, o \rangle$ tuples that reflecting the world state, and a set of *valid actions*. Valid actions in Jericho

Prev. Action: *drop torch*		
Observation	**Graph**	**Valid actions**
Dropped. Location: Great Door. This is the south end of a monumental hall, full of dust and debris from a recent earthquake. To the east is a great iron door, rusted shut. To its right, however, is a gaping cleft in the rock and behind, a cleared area. There is a torch here. You are carrying: • A cloak (being worn) • A hood (being worn) • A vial • A wooden staff • A strange key • A Frobozz Magic Grue Repellent • A golden amulet (being worn)	'you in Great Door', 'you have key', 'you have vial', 'you have hood', 'torch in Great Door', 'you have golden amulet', 'you have Frobozz Magic Grue Repellent', 'you have cloak', 'heavy invisible liquid in vial', 'you have staff'	• blow out torch • take torch • put down magic • put down key • put down staff • put down amulet • put down vial • put down hood • put down cloak • put down all • open vial • light magic with torch • light staff with torch • east • north
Action: *put down amulet*		
True positives	**False positives**	**False negatives**
• light magic with torch • blow out torch • take amulet • east • take all • open vial	• push amulet to ground	• take torch • north • put down key • put down magic • put down staff • put down all • put down hood • put down cloak • put down vial • light staff with torch

Table 1: Illustration of the inputs and generated outputs of the valid action prediction task, on *Zork3*. The results are from a Worldformer-BART model.

are actions recognized by the game's parser that cause changes in the world state.

Overall, there are 24,198 training instances from 27 text games in multiple genres, and 7,836 heldout test instances from 9 additional games. Notably, the test set consists of a diverse set of never-before-seen text games, requiring zero-shot prediction of world modeling tasks. As the test games differ widely in terms of genre and structure, and are never seen during training, we follow the convention in the literature to refer to these games as out-of-distribution games (Adhikari et al., 2020; Atzeni et al., 2021).

2.2.1 JerichoWorld Tasks

A state S_t in each example of JerichoWorld consists of $O_t, V_t, G_t \in S_t$, indicating the textual observation, the valid actions, and the knowledge graph, respectively.[2] JerichoWorld proposes two predictive world modeling tasks:

The first task is to predict the future graph at time step $t+1$, $G_{t+1} \in S_{t+1}$, from $O_t, V_t, G_t \in S_t$ and the transition action A_t. For the graph prediction task, we follow the simplification in Ammanabrolu and Riedl (2021a), to limit the task to predicting node additions, i.e. predicting the graph difference caused by additions, as these are sufficient to infer node deletions as well.

The second task is to predict the set of future valid actions at time step $t+1$, $V_{t+1} \in S_{t+1}$, from $O_t, V_t, G_t \in S_t$ and the transition action A_t. In this work, we focus on the valid action prediction task, which we illustrate in detail in Table 1.

2.2.2 JerichoWorld 2.0

Since JerichoWorld is generated by simulating Jericho games, to ensure there are no data artifact issues, we also provide an updated version of JerichoWorld. We follow the methodology in Ammanabrolu and Riedl (2021b) to generate the data, making sure to generate roughly the same number of instances for each game, from the same human walkthroughs. Overall, there are 24,198 training instances and 7505 test instances in our dataset. A comparison of the test sets can be found in Appendix A.4. Our dataset additionally provides templates and objects for all actions, as well as de-abbreviating commands from the human playthrough for consistency of commands.

3 Preliminaries

We now describe our base models in detail. We begin with a description of Worldformer, which forms the basis of our model architectures.

3.1 Background: Worldformer

Worldformer is a multi-task architecture designed to perform the dual world modeling tasks of JerichoWorld. The model consists of two BERT-based encoders and two randomly initialized transformer decoders. The first pair of encoder and decoder solves the future valid action prediction task, and the second solves the future graph prediction task. Both tasks are learned simultaneously via multi-task learning. Additionally, a domain-adaptive MLM task is used to further pre-train the encoders, and a domain-specific vocabulary and tokenizer built for Jericho is used for both encoders and decoders. Ammanabrolu and Riedl (2021a) show that Worldformer achieves state-of-the art performance on both tasks.

[2]The textual observation consists of the engine response to the previous action, and the engine response to the commands, "*look*", and "*inventory*".

3.2 Baseline: Worldformer-BART

As a starting point for adapting PLMs as world models, we build a world model based on adapting BART with minimal changes. The model consists of two pre-trained BART models, arranged in a similar multi-task architecture to Worldformer. We finetune both BARTs directly, without any further pre-training or any changes to vocabulary or tokenization, such as building a domain-specific vocabulary as in Ammanabrolu and Riedl (2021a), or limiting the softmax operation. We give a full description of the model below. For ease of notation, we divide BART into its encoder and decoder components, where $\mathsf{BART}^{enc}_{text}$ and $\mathsf{BART}^{dec}_{action}$ together compose the first BART encoder-decoder, and $\mathsf{BART}^{enc}_{graph}$ and $\mathsf{BART}^{dec}_{graph}$ compose the second.

Formally, the JerichoWorld example $\langle S_t, A_t, S_{t+1}, R\rangle$ yields two sets of input and target token sequences, $X = \{x_1, ..., x_n\}$ and $Y = \{y_1, ..., y_n\}$. That is, for the valid action prediction task, $\langle X, Y\rangle = \langle O_t +\!\!+ V_t +\!\!+ A_t, V_{t+1}\rangle$, and for the graph prediction task, $\langle X, Y\rangle = \langle G_t +\!\!+ A_t, G_{t+1}\rangle$ where $+\!\!+$ indicates string concatenation. During model operation, each BART encoder produces contextual encodings given their respective inputs, $\mathbf{O}_t = \mathsf{BART}^{enc}_{text}(O_t +\!\!+ V_t +\!\!+ A_t)$ and $\mathbf{G}_t = \mathsf{BART}^{enc}_{graph}(G_t +\!\!+ A_t)$. As in Worldformer, the aggregation module produces the state vector, $s_t = \mathsf{Agg}(\mathbf{O}_t, \mathbf{G}_t)$

Without loss of generality, both tasks are modeled by the conditional distribution,

$$P(Y|X) = \prod_{i=1}^{n} P(y_i|y_{<i-1}, s_t; \mathsf{BART}^{enc}(X)) \tag{1}$$

and each BART encoder-decoder is trained to generate the target sequence through maximum log likelihood loss, as follows:

$$\begin{aligned}\mathcal{L}_{gen} &= \log P(Y|X) \\ &= \sum_{i=1}^{n} \log p(y_i|y_{<i-1}, s_t; \mathsf{BART}^{enc}(X))\end{aligned} \tag{2}$$

where p is modeled by corresponding BART^{dec}.

3.3 Motivation: Qualitative Study

To motivate actionability objectives, we qualitatively analyze Worldformer-BART, which is an effective starting point for building actionable world models, but with three major types of actionability errors:

- *Object Localization and Inference (OLI) errors*: We define these as reasoning errors wherein the model fails to track the current location of object(s), i.e. whether an object is found in the inventory, the surrounding environment, or is not found at all. Examples include attempting to put a first aid kit down, before ever having picked it up, or trying to open a case of cigarettes after it is no longer in the player's possession. These constitute false positive generated actions.
- *Object Affordance errors*: These errors occur when an incorrect understanding of the affordance of objects leads the model to generate actions which are nonsensical or impossible. Examples include asking a library about a library, attempting to drink out of an empty bucket, or looking with a net, etc. These also constitute false positive generated actions.
- *Insufficient Interaction Coverage*: We define these errors as those in which, despite the presence of objects, the model's generation is insufficient to enumerate all possible interactions with the objects. These errors can be caused by errors of reasoning, or by the inability of decoding schemes to generate with high coverage. Most false negatives fall into this category.

We perform human analysis of 50 randomly sampled examples from the validation set. A subset of samples from the human analysis can be found in Appendix A.7. Of the analyzed samples, we find that ~33%, ~48%, and ~88% of samples exhibit each type of error, respectively. These results indicate that actionability for world models is not sufficiently satisfied by naive adaptation of PLMs.

4 AWM-BART for Actionable World Model

We now propose our model, which aims to improve the actionability of the BART-based world model. We decompose action generation as template selection and filling, aiming to capture two benefits: Through input-constrained decoding using templates, we enhance the parseability of the world model's generations. Furthermore, we posit that the inductive bias from templates will aid the world model to learn more accurate object affordance and object localization, improving commonsense.

More specifically, $\text{BART}^{dec}_{action}$ now performs the task of generating the sequence of possible fillings $Y^{t_j} = \{y_1^{t_j}, ..., y_n^{t_j}\}$, $Y = \{Y^{t_1}, ..., Y^{t_m}\}$ of a template $t_j \in T_{env}$, of a Jericho game environment. Then, the template-conditional action generation task is reformulated as follows:

$$P(Y^{t_j}|X, t_j) = \prod_{i=1}^{n} P(y_i^{t_j}|y_{<i-1}^{t_j}, s_t; \text{BART}^{enc}_{text}(O_t +\!\!+ V_t +\!\!+ A_t +\!\!+ t_j)) \quad (3)$$

The mask filling task is illustrated in Fig.1. The above formulation is advantageous in that the task of filling the masks of an action template brings the generation objective close to BART's original objective. However, it does not address the issue of *choosing* an appropriate template to fill. We next describe how we employ multitask learning to utilize a single BART encoder-decoder as both a template retrieval and generation model.

4.1 Template Retrieval

In contrast to previous works utilizing template selection and filling (Hausknecht et al., 2019; Ammanabrolu and Hausknecht, 2020) for in-domain learning of agents, our aim is to build a model to generalize to any arbitrary set of natural language templates, as the world modeling tasks of JerichoWorld require zero-shot prediction on unseen games. Taking inspiration from recent advances in retrieval using neural models, we propose to extend BART^{enc}_{text} as a retriever. The goal of the retriever is to identify the subset of templates $T^{valid}_{t+1} \subseteq T_{env}$, i.e. the templates defining the valid actions V_{t+1}, given the current world state S_t and the transition action A_t. Therefore, we use the set of future valid actions, V_{t+1}, to extract the valid templates T^{valid}_{t+1}.

In the retrieval nomenclature, first and second-stage retrieval refer respectively to a fast and efficient retrieval model to quickly identify a set of promising candidates, and a computationally expensive but effective *re-ranking* model, which produces fine-grained rankings over the smaller set of first-stage candidates. In our case, it is possible to adopt BART^{enc}_{text} as a dense first-stage retriever (Lee et al., 2019; Karpukhin et al., 2020), as well as a re-ranker (Nogueira et al., 2019). Note that in our setting, since we have access to the set of all possible templates, and their number does not exceed 300 for any environment, we forego the usage of a first-stage retriever, and simply enumerate over all templates (on average around ~200).

To learn the relevance score r_j between $\langle O_t, V_t, A_t \rangle$ and each template $t_j \in T_{env}$, for each template t_j we concatenate t_j to the encoder input. The BART encoder operates as before, but we now extract a summary vector as well as the contextual encodings, i.e. $(\mathbf{O}_t, o_t) = \text{BART}^{enc}_{text}(O_t +\!\!+ V_t +\!\!+ A_t +\!\!+ t_j)$. Here, o_t can be any pooled vector, and we choose the vector encoding of the EOS token. Similarly, we extract g_t as $(\mathbf{G}_t, g_t) = \text{BART}^{enc}_{graph}(O_t +\!\!+ V_t +\!\!+ A_t)$. These vectors are concatenated with state vector s_t, and fed to a re-ranking head, which computes the template relevance score r_j:

$$r_j = P(l_j = 1; o_t \oplus g_t \oplus s_t) \quad (4)$$

where l_j indicates the ground truth label of template t_j. The re-ranking cross-entropy loss is defined as follows:

$$\mathcal{L}_{rerank} = -\sum_{i \in I_{\text{pos}}} \log(r_i) - \sum_{i \in I_{\text{neg}}} \log(1 - r_i) \quad (5)$$

where I_{pos} are the indices of templates in T^{valid}_{t+1}, and I_{neg} are the indices of templates belonging to the complement set. The re-ranking objective enables the full utilization of the BART encoder to model fine-grained relationships between the input context and each template.

4.2 Template Filling

In addition to the template retrieval task, given template t_j, we use $\text{BART}^{dec}_{action}$ to generate the filled version of the template, with P defined in Eq. 3.

$$\mathcal{L}^{action}_{fill} = \log P(Y^{t_j}|X, t_j) \quad (6)$$

As per the definition in Eq 3, the same BART^{enc}_{text} is shared between both retrieval and filling tasks, allowing efficient adaptation of BART to the target controlled sublanguage through multitask learning. We observe that conditioning the encoder alone with templates can effectively force the decoder to produce faithful fillings of the provided masked template. While template constraints intuitively improve the actionabillity of model generations in terms of parseability, we additionally expect that templates can provide a useful inductive bias for reducing OLI and Object Affordance errors. Our hypothesis is that, since the decoder is trained to fill only a single masked template at a time, this

has the implicit effect of marginalizing out the effect of other templates. To see why, consider the default decoding objective which treats all valid actions (generated from all $t_j \in T_{t+1}^{valid}$) as a single sequence to be generated. This causes every action to be conditioned on other valid templates $t_j \in T_{t+1}^{valid}$ which appear at a previous position in the target sequence. In contrast, our template conditioned objective removes this effect, replacing it with a strong conditioning on the masked template, allowing the affordance relationship between template verbs and their corresponding objects to be learned efficiently. Finally, during inference time, the template retrieval module works as an effective filter which refines the action space to a promising subset, further reducing the room for error.

4.3 Training

In our experiments, we found it most effective to train the model in phases. In the first phase, we multi-task train the template filling task for action generation, together with the graph prediction task:

$$\mathcal{L}_{phase1} = \mathcal{L}_{fill}^{action} + \mathcal{L}_{gen}^{graph} \quad (7)$$

Then, using the trained weights, we train on all losses simultaneously in the second stage:

$$\mathcal{L}_{phase2} = \mathcal{L}_{fill}^{action} + \mathcal{L}_{gen}^{graph} + \mathcal{L}_{rerank} \quad (8)$$

Note that, when training in the second phase, batch items vary depending on whether the template t_j in the input $O_t +\!\!+ V_t +\!\!+ A_t +\!\!+ t_i$ has label $l_j = 1$, or $l_j = 0$. We activate the full loss only in the former case, and only activate $\mathcal{L}_{rerank}$ in the latter via a loss mask.

4.4 Hard Negatives Mining

Our retrieval formulation motivates our application of the technique of hard negative example mining to the template retrieval task. We supply hard negatives from a model trained with Eq. 8, as additional negative examples for Eq. 5. As we later show, hard negatives further improve the accuracy of reranking, and the overall performance of valid action generation.

5 Experiments

We evaluate our models on the JerichoWorld modeling tasks.

5.1 Metrics

We report the F1 and EM metrics from Ammanabrolu and Riedl (2021b), where F1 is a harmonic mean of predicted precision and recall, while EM (exact match) checks for accuracy or direct overlap between the predictions and ground truth. The original dataset defines F1 and EM at two different levels: token-level, and graph tuple-level.

We focus on the tuple-level metrics, as they are the main metrics for the action task in Ammanabrolu and Riedl (2021a) . For the graph task, a tuple-level true positive occurs when all three items within an $\langle s, r, o \rangle$ tuple[3] matches a tuple within the ground truth graph. The same holds for the action task, where predicted and ground truth valid actions are likewise defined as ordered tuples of tokens. The tuple-level metrics are stricter and more relevant for actionability, as EM match means the model generated an action correctly in its entirety, making the action executable by the game engine.

5.2 Baselines

We compare AWM-BART with the following models:

1. *Worldformer-BART Action decoder from scratch* is our reimplementation of Worldformer (Ammanabrolu and Riedl, 2021a). To reproduce Worldformer, which is a multi-task world model composed of pre-trained BERT encoders and transformer decoders, We initialized the action decoder weights from scratch, while keeping the rest of the BART pre-trained weights. Under such minor modification, we achieved similar results reported in original Worldformer.[4]

2. *CALM* (Yao et al., 2020) is a GPT-2 based model finetuned to generate V_{t+1} from O_t, A_t, O_{t+1}. While it does not directly train on the Jericho suite of games, it trains on on a dataset of 426 human gameplay transcripts for 590 different text-based games, ClubFloyd[5].

3. *Worldformer-BART* is our implementation of Worldformer based on BART, a simple adaptation of PLMs as world models.

4. *Worldformer-BART + COMET*[6] is a COMET-augmented version of Worldformer-BART. We use logits of a pre-trained COMET model to filter out

[3] We do not use separator tokens within each $\langle s, r, o \rangle$ tuple in the graph generation task.

[4] Original Worldformer results can be found in Appendix A.3

[5] http://www.allthingsjacq.com/interactive_fiction.html

[6] Implementation details in Appendix A.5

Models	Task	Game	zork1	lib.	det.	bal.	pent.	ztuu	ludi.	deep.	temp.	overall
CALM (k=10)	**Action**	**EM**	14.15	11.84	26.98	5.19	17.23	6.46	10.73	7.64	10.62	11.07
		F1	24.04	20.51	41.53	9.55	27.90	11.56	18.78	13.55	18.55	18.98
Worldformer-BART	**Graph**	**EM**	34.76	24.38	37.92	41.96	36.51	40.92	48.54	20.52	45.09	39.95
		F1	35.35	25.18	39.00	43.02	37.54	41.62	48.77	20.67	46.07	40.59
Action decoder from scratch	**Action**	**EM**	11.59	21.45	32.25	5.61	24.07	6.35	12.81	9.42	13.68	13.44
		F1	19.47	32.77	46.72	10.06	35.53	11.04	21.25	16.40	22.36	21.69
Worldformer-BART	**Graph**	**EM**	35.42	24.15	39.99	36.55	34.62	41.75	52.48	21.35	43.75	40.38
		F1	36.05	24.69	41.01	37.44	35.70	42.51	52.97	21.51	45.09	41.12
	Action	**EM**	27.04	43.79	44.22	15.69	40.02	16.00	22.27	19.30	26.03	25.56
		F1	40.46	56.31	57.53	26.08	52.91	25.79	34.13	30.80	38.52	37.51
Worldformer-BART	**Graph**	**EM**	35.42	24.15	39.99	36.55	34.62	41.75	52.48	21.35	43.75	40.38
		F1	36.05	24.69	41.01	37.44	35.70	42.51	52.97	21.51	45.09	41.12
+ COMET filter	**Action**	**EM**	27.46	43.72	44.64	15.89	41.28	16.14	22.71	19.42	25.71	25.78
		F1	40.99	56.16	57.89	26.44	54.03	26.05	34.69	31.06	38.04	37.78
AWM-BART	**Graph**	**EM**	35.46	27.41	38.78	43.64	37.17	43.60	50.96	22.42	46.87	***41.87**
		F1	36.03	28.08	39.63	44.68	38.10	44.35	51.29	22.54	47.99	***42.53**
	Action	**EM**	43.21	58.46	51.20	38.16	50.48	39.12	34.63	35.40	40.15	***40.67**
		F1	58.29	70.09	64.10	53.97	63.39	53.63	49.35	51.02	54.55	***55.16**

Table 2: Results on JerichoWorld world modeling tasks. We report experimental results on JerichoWorld 2.0. Best overall results are denoted in bold. Asterisk (*) denotes statistically significant (P<.001) improvement over Worldformer-BART using a paired t-test. All results are averaged over three random seeds, with standard deviation under ± 1.60 in the overall categories for either task.

actions, based on the *commonsenseness* of actions given the current observation and inventory.

6 Results

We report the results of our experiments in Table 2. Our Worldformer-BART performs on par with Worldformer on graph prediction, but shows improvement on action generation, due to the leveraging of BART. Augmenting Worldformer-BART with a COMET-based filter fails to improve the action generation performance meaningfully, indicating that zero-shot adaptation of conventional commonsense PLMs to TBGs is challenging. Compared to CALM, our model is significantly better in action generation, indicating the importance of considering actionability in adapting PLMs to TBGs. Our model was able to outperform all compared models in both tasks, achieving a significant improvement in action generation.

6.1 Ablation Study

In order to validate the usefulness of each of our model components, we report the results of the ablation study in Table 3. We begin by comparing Worldformer-BART and AWM-BART trained with Eq. 8 without hard negatives (AWM-BART - hard negatives), where we observe that our proposed template-constrained architecture improves the learning of both tasks over Worldformer-BART. We next compare the two AWM-BART variants with and without hard negatives, which shows that while maintaining graph prediction performance, hard negatives significantly boost valid action generation, by making the reranking head more robust and reducing false positives. Finally, we report the results from using an oracle template retriever with AWM-BART (AWM-BART + Oracle). We can see that our trained template retriever approaches the performance of the oracle retriever, which indicates that there are potentially more improvements to be gained by improving the decoder.

6.2 Commonsense Study

In order to scale the analysis from Sec. 3.3 to the entire test set, we build an automated, rule-based system for detecting the OLI and affordance errors. We validate the system on the human-annotated samples, where the system recovered 91% of the human-annotated OLI and affordance errors. Implementation details are provided in Appendix A.6. In Table 4, we report the results of the commonsense error analysis using the automated system. The purpose of this study is to determine whether the improved performance of AWM-BART is attributable to the model's improved commonsense understanding. We compare our model with Worldformer-BART, and Worldformer-BART with a COMET filter.

We find that the number of errors successfully filtered out by COMET was negligible, and was outpaced by the increase in the number of false negatives. In contrast, relative to Worldformer-BART, AWM-BART reduces OLI errors by ~47%, affordance errors by ~33%, and insufficient generation coverage errors by ~24%. Taken in conjunction with the results on world modeling tasks,

Models	Task	Game	zork1	lib.	det.	bal.	pent.	ztuu	ludi.	deep.	temp.	overall
AWM-BART	**Graph**	**EM**	35.46	27.41	38.78	43.64	37.17	43.60	50.96	22.42	46.87	**41.87**
		F1	36.03	28.08	39.63	44.68	38.10	44.35	51.29	22.54	47.99	**42.53**
+ Oracle retriever	**Action**	**EM**	53.69	71.27	73.10	42.71	64.13	45.11	40.57	40.82	48.17	**48.94**
		F1	67.74	80.12	82.40	58.53	74.46	59.44	55.01	56.52	62.09	**62.63**
AWM-BART	**Graph**	**EM**	35.46	27.41	38.78	43.64	37.17	43.60	50.96	22.42	46.87	**41.87**
		F1	36.03	28.08	39.63	44.68	38.10	44.35	51.29	22.54	47.99	**42.53**
	Action	**EM**	43.21	58.46	51.20	38.16	50.48	39.12	34.63	35.40	40.15	40.67
		F1	58.29	70.09	64.10	53.97	63.39	53.63	49.35	51.02	54.55	55.16
AWM-BART	**Graph**	**EM**	35.62	25.92	38.91	44.06	38.05	42.85	50.80	22.08	47.48	41.83
		F1	36.20	26.69	39.89	45.10	39.03	43.53	51.11	22.19	48.60	42.50
- hard negatives	**Action**	**EM**	31.99	31.92	25.87	33.47	29.38	34.02	24.92	30.33	28.07	29.04
		F1	46.66	44.26	37.90	48.85	41.99	48.54	37.91	45.24	41.61	42.73
Worldformer-BART	**Graph**	**EM**	35.42	24.15	39.99	36.55	34.62	41.75	52.48	21.35	43.75	40.38
		F1	36.05	24.69	41.01	37.44	35.70	42.51	52.97	21.51	45.09	41.12
	Action	**EM**	27.04	43.79	44.22	15.69	40.02	16.00	22.27	19.30	26.03	25.56
		F1	40.46	56.31	57.53	26.08	52.91	25.79	34.13	30.80	38.52	37.51

Table 3: Results of ablation experiments. All results are averaged over three random seeds, with standard deviation under ± 1.60 in the overall categories for either task.

Models	Error Types		
	False Positive		False Negative
Worldformer-BART	20803		88317
Ours	17299		67545
	OLI	**Affordance**	**Insufficient**
Worldformer-BART	8066	3624	88317
Worldformer-BART + COMET	7857 (-2.60%)	3609 (-0.41%)	88802 (+0.55%)
AWM-BART	4260 (-47.18%)	2433 (-32.86%)	67536 (-23.79%)

Table 4: Number of errors per type for compared models as measured by automated system. Parentheses indicate the percentage reduction of each type of error, with respect to Worldformer-BART. All results are averaged over three random seeds.

these findings lend support to our hypothesis that the template-constrained generation of our model enhances not only the parseability in a target controlled sublanguage, but improves the commonsense understanding of the PLM as a world model. While the results of AWM-BART are encouraging, our findings raise new questions about precisely what kind of commonsense is being captured by PLMs, and whether the conventional definition of commonsense in the literature is general enough to capture the varied ways in which humans can employ commonsense, as they do in TBGs.

7 Related Works

7.1 Template-based Action Space

For in-domain generalization to a single-game setting, previous works such as LSTM-DQN (Narasimhan et al., 2015), TDQN (Hausknecht et al., 2019), and KG-A2C (Ammanabrolu and Hausknecht, 2020) have proposed using template-based action spaces in text-based games. Our distinction of employing templates, is to improve actionability of PLM-based world models. In particular, we design a template retrieval task which allows our model to generalize to any arbitrary set of templates, unlike previous works. Template retrieval plays a key role in ours, for generalizing to unseen, out-of-distribution games.

7.2 PLMs for TBG

While research adapting PLMs for TBGs is still in its early stages, there are notable works which have leveraged the linguistic, semantic and commonsense priors of PLMs to TBG agents. DBERT-DRRN (Singh et al., 2022) proposes a single-game agent based on DistilBERT (Sanh et al., 2019) to improve the semantic understanding of agents, achieving state of the art on performance on several games in Jericho. CALM (Yao et al., 2020) trains a GPT-2 model to train a single model which can be deployed to generate actions across many different downstream games, and CALM is shown to improve the existing agents on unseen games by reranking action candidates to maximize rewards. Like CALM, we tackle generalizing PLMs to unseen games, but our distinction is to consider actionability as the key criterion in doing so. Our results show that actionability is indeed crucial in building PLM-based world models for TBGs.

7.3 Commonsense in Text-based Games

Prior works have proposed incorporating external commonsense knowledge to TBG agents. Dambekodi et al. (2020); Ryu et al. (2022) propose to incorporate commonsense knowledge into a KG-A2C agent by augmenting its graph with commonsense inferences using COMET (Bosselut et al., 2019). Murugesan et al. (2021) propose to

jointly leverage a commonsense knowledge graph directly retrieved from ConceptNet along with a textual graph. Ammanabrolu and Riedl (2019) propose to use knowledge graphs to transfer commonsense across agents. Our distinction is to envision and enhance commonsense as an innate component of TBG world models.

8 Conclusion

In this work, we build world models based on PLMs. Towards leveraging semantic and linguistic priors learned by PLMs, we identified major areas in which PLMs need to be systematically improved, namely generation to a controlled sublanguage, and commonsense understanding. We propose an actionable world model based on template-augmented generation, showing that both parseability and commonsense understanding can be significantly improved. As future work, we consider combining PLM-based world models with reward-based agents to learn goal-directed policies as a promising direction. Finally, our hope is that our work will contribute to the active exploration of text-based games as an alternative testbed for further research on commonsense reasoning.

Limitations

A limitation of our work is that while we aim to adapt pre-trained language models as world models for text-based game environments, due to limited computational resources, we are not able to test our models on larger scales, and take greater advantage of language model scaling laws (Kaplan et al., 2020). As such, our results should be understood in terms of the intermediate-scale regime of PLMs. Nevertheless, we believe that actionable world models should keep efficiency as one of their core criteria.

A second limitation is that in this work, our experiments have focused solely on data which originates from the Jericho suite of games. While this is outside the immediate scope of this work, there are several other TBG environments which may be suitable candidates for learning PLM-based world models. While scaling to a greater number of TBG environments is a challenging task, we hold the view that the promise of world models in TBGs will be fulfilled by models which generalize to many environments across varying domains, structures and rules.

Acknowledgements

This work was partly supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government(MSIT) [NO.2021-0-01343, Artificial Intelligence Graduate School Program (Seoul National University)] and [(No. 2022-0-00077, AI Technology Development for Commonsense Extraction, Reasoning, and Inference from Heterogeneous Data)].

References

Ashutosh Adhikari, Xingdi Yuan, Marc-Alexandre Côté, Mikuláš Zelinka, Marc-Antoine Rondeau, Romain Laroche, Pascal Poupart, Jian Tang, Adam Trischler, and Will Hamilton. 2020. Learning dynamic belief graphs to generalize on text-based games. In *Advances in Neural Information Processing Systems*, volume 33, pages 3045–3057. Curran Associates, Inc.

Alan Akbik, Duncan Blythe, and Roland Vollgraf. 2018. Contextual string embeddings for sequence labeling. In *COLING 2018, 27th International Conference on Computational Linguistics*, pages 1638–1649.

Prithviraj Ammanabrolu and Matthew Hausknecht. 2020. Graph constrained reinforcement learning for natural language action spaces. In *International Conference on Learning Representations*.

Prithviraj Ammanabrolu and Mark Riedl. 2019. Transfer in deep reinforcement learning using knowledge graphs. In *Proceedings of the Thirteenth Workshop on Graph-Based Methods for Natural Language Processing (TextGraphs-13)*, pages 1–10, Hong Kong. Association for Computational Linguistics.

Prithviraj Ammanabrolu and Mark O. Riedl. 2021a. Learning knowledge graph-based world models of textual environments. In *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pages 3720–3731.

Prithviraj Ammanabrolu and Mark O. Riedl. 2021b. Modeling worlds in text. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks 1, NeurIPS Datasets and Benchmarks 2021, December 2021, virtual*.

Mattia Atzeni, Shehzaad Dhuliawala, Keerthiram Murugesan, and Mrinmaya Sachan. 2021. Case-based reasoning for better generalization in text-adventure games. *CoRR*, abs/2110.08470.

Antoine Bosselut, Ronan Le Bras, , and Yejin Choi. 2021. Dynamic neuro-symbolic knowledge graph construction for zero-shot commonsense question

answering. In *Proceedings of the 35th AAAI Conference on Artificial Intelligence (AAAI)*.

Antoine Bosselut, Hannah Rashkin, Maarten Sap, Chaitanya Malaviya, Asli Celikyilmaz, and Yejin Choi. 2019. COMET: Commonsense transformers for automatic knowledge graph construction. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4762–4779, Florence, Italy. Association for Computational Linguistics.

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.

Sahith N. Dambekodi, Spencer Frazier, Prithviraj Ammanabrolu, and Mark O. Riedl. 2020. Playing text-based games with common sense. *CoRR*, abs/2012.02757.

David Ha and Jürgen Schmidhuber. 2018. Recurrent world models facilitate policy evolution. In *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc.

Danijar Hafner, Timothy P. Lillicrap, Mohammad Norouzi, and Jimmy Ba. 2020. Mastering atari with discrete world models. *CoRR*, abs/2010.02193.

Matthew Hausknecht, Prithviraj Ammanabrolu, Côté Marc-Alexandre, and Yuan Xingdi. 2019. Interactive fiction games: A colossal adventure. *CoRR*, abs/1909.05398.

Jena D. Hwang, Chandra Bhagavatula, Ronan Le Bras, Jeff Da, Keisuke Sakaguchi, Antoine Bosselut, and Yejin Choi. 2021. Comet-atomic 2020: On symbolic and neural commonsense knowledge graphs. In *AAAI*.

Lukasz Kaiser, Mohammad Babaeizadeh, Piotr Milos, Blazej Osinski, Roy H. Campbell, Konrad Czechowski, Dumitru Erhan, Chelsea Finn, Piotr Kozakowski, Sergey Levine, Ryan Sepassi, George Tucker, and Henryk Michalewski. 2019. Model-based reinforcement learning for atari. *CoRR*, abs/1903.00374.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. 2020. Scaling laws for neural language models. *CoRR*, abs/2001.08361.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6769–6781, Online. Association for Computational Linguistics.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Kenton Lee, Ming-Wei Chang, and Kristina Toutanova. 2019. Latent retrieval for weakly supervised open domain question answering. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6086–6096, Florence, Italy. Association for Computational Linguistics.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

Keerthiram Murugesan, Mattia Atzeni, Pavan Kapanipathi, Kartik Talamadupula, Mrinmaya Sachan, and Murray Campbell. 2021. Efficient text-based reinforcement learning by jointly leveraging state and commonsense graph representations. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 719–725, Online. Association for Computational Linguistics.

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, Lisbon, Portugal. Association for Computational Linguistics.

Rodrigo Frassetto Nogueira, Wei Yang, Kyunghyun Cho, and Jimmy Lin. 2019. Multi-stage document ranking with BERT. *CoRR*, abs/1910.14424.

Dongwon Ryu, Ehsan Shareghi, Meng Fang, Yunqiu Xu, Shirui Pan, and Reza Haf. 2022. Fire burns, sword cuts: Commonsense inductive bias for exploration in text-based games. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 515–522, Dublin, Ireland. Association for Computational Linguistics.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of BERT: smaller, faster, cheaper and lighter. *CoRR*, abs/1910.01108.

Richard Shin, Christopher Lin, Sam Thomson, Charles Chen, Subhro Roy, Emmanouil Antonios Platanios, Adam Pauls, Dan Klein, Jason Eisner, and Benjamin Van Durme. 2021. Constrained language models yield few-shot semantic parsers. In *Proceedings of*

the 2021 Conference on Empirical Methods in Natural Language Processing, pages 7699–7715, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Ishika Singh, Gargi Singh, and Ashutosh Modi. 2022. Pre-trained language models as prior knowledge for playing text-based games. In *21st International Conference on Autonomous Agents and Multiagent Systems, AAMAS 2022, Auckland, New Zealand, May 9-13, 2022*, pages 1729–1731. International Foundation for Autonomous Agents and Multiagent Systems (IFAAMAS).

Jack Urbanek, Angela Fan, Siddharth Karamcheti, Saachi Jain, Samuel Humeau, Emily Dinan, Tim Rocktäschel, Douwe Kiela, Arthur Szlam, and Jason Weston. 2019. Learning to speak and act in a fantasy text adventure game. 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 673–683, Hong Kong, China. Association for Computational Linguistics.

Shunyu Yao, Rohan Rao, Matthew Hausknecht, and Karthik Narasimhan. 2020. Keep CALM and explore: Language models for action generation in text-based games. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8736–8754, Online. Association for Computational Linguistics.

Denis Yarats, Amy Zhang, Ilya Kostrikov, Brandon Amos, Joelle Pineau, and Rob Fergus. 2021. Improving sample efficiency in model-free reinforcement learning from images. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(12):10674–10681.

A Appendix

A.1 Model Architecture details

We use BART-base (Lewis et al., 2020) to build Worldformer-BART and AWM-BART. The overall parameters counts of the models are 320 million and 322 million, respectively, indicating a roughly ~15% reduction relative to Worldformer, which has 380 million parameters. We use BART's default tokenizer with the exception of adding several special tokens for usage as delimiters. All models generate with beam search decoding, using a beam size of 15.

A.2 Training details

We use Adam optimizer (Kingma and Ba, 2015), with learning rate 3×10^{-5} and batch size 16 to train our models. For AWM-BART, we first train with Eq. 7 for a single epoch, then continue training with Eq. 8, reducing learning rate by $1/10$. For the latter phase, to train the retrieval head, we randomly sample negative templates with 1:1 ratio between positive and negative templates. When applying additional hard negative mining, the mined templates are added to the pool of randomly sampled negatives, increasing the total number of negative templates. We perform model selection based on the combined loss of all tasks on the validation set. Our models are trained using a single NVIDIA GeForce RTX 3090 Ti GPU, taking between 3 and 5 hours per epoch.

A.3 Original Worldformer Results

As a reference, report the results reported in the original Worldformer, on the original JerichoWorld dataset in Table 5.

A.4 JerichoWorld 2.0 Test set details

In Table 6, we compare the the test sets of JerichoWorld 2.0 with that of the original, to confirm that the distribution of instances across of games remains consistent.

A.5 Implementation COMET-based Score Filter

We adopt an off-the-shelf COMET-BART model (Hwang et al., 2021) as a commonsenseness scorer. We follow the method in Bosselut et al. (2021); Ryu et al. (2022) to compute the score of a target sequence of tokens, based on COMET's logits. We feed the current observation context and inventory, concatenated with the transition action and an ATOMIC2020 relation as the input to COMET, as done in Ryu et al. (2022). Each token in the target sequence, which is an action being evaluated, is then fed sequentially to COMET, to obtain model logits. We use the relations xNeed and xWant, averaging the two scores for each token, and the overall score of the action is given as the average over the tokens in the action. To filter out actions, we compute the mean and standard deviation of scores of all generated actions per data example, and drop outliers whose scores are more than 1 standard deviation lower than the mean.

A.6 Implementation of Automated Commonsense Error Detector

Based on our preliminary analysis, we develop a heuristic method for detecting commonsense errors defined in Sec. 3.3. We leverage the $\langle S_t, A_t, S_{t+1} \rangle$ tuples in the dataset, to directly measure these er-

Expt.	Task	Game	zork1	lib.	det.	bal.	pent.	ztuu	ludi.	deep.	temp.	overall
		Size	886	654	434	990	276	462	2210	630	1294	7836
Worldformer	Graph	EM	21.62	34.39	41.05	50.41	30.00	41.56	40.10	41.87	42.43	39.15
		F1	24.44	34.39	44.53	52.43	34.30	42.20	41.65	42.74	45.17	41.06
	Action	EM	23.08	22.55	20.97	29.08	27.05	20.71	21.36	24.04	22.80	23.22
		F1	23.50	26.52	25.28	32.89	31.32	23.66	22.27	26.12	25.66	25.54

Table 5: Original Worldformer Results on JerichoWorld.

JerichoWorld

Game	zork1	lib.	det.	bal.	pent.	ztuu	ludi.	deep.	temp.	overall
Size	886	654	434	990	276	462	2210	630	1294	7836

JerichoWorld 2.0

Game	zork	lib.	det.	bal.	pent.	ztuu	ludi.	deep.	temp.	overall
Size	779	624	417	971	266	441	2170	608	1229	7505

Table 6: Comparison of the test sets of JerichoWorld and JerichoWorld 2.0

rors automatically. Specifically, we detect the following types of errors:

- **Object Localization and Inference errors**: For false positive actions, 1. Predicted valid action attempts to interact with an object not present in the current environment or inventory. 2. Predicted valid action attempts to take object from the environment when it is expected to in inventory already, after the transition action. 3. Predicted valid action attempts to put object down when object is not expected to be in inventory after the transition action. Furthermore, we build special heuristics for when the object in question is 'all', as the game's parser understands this as referring to all objects in either the environment or the player inventory, depending on the action.
- **Object Affordance Errors**: After first filtering for OLI errors, we check an action for the following: First, we check that a false positive action has at least two noun phrases, such that an affordance between two objects can be established. Next, to account for idiosyncrasies of the engine in terms of action validity (e.g. *push lens to glasses* is a valid action), we check that the action was not a valid action in the previous state. If the action passes both checks, we judge the action to be an affordance error (e.g. *throw lantern at garlic*)
- **Insufficient Iteraction Coverage**: We consider all false negatives as these.

To extract objects in predicted actions, we extract noun phrases using the Flair framework (Akbik et al., 2018). To check the existence of objects in the environment, inventory and graph, we use simple string matching of the object name.

A.7 Qualitative study samples

In Tables 7 through 9, we show a subset of the dev set error analysis results, comparing the human annotated errors with the annotation by our automated system.

A.8 Examples of model predictions

In Tables 10 through 14, we show several examples of models' valid action predictions on the test set. Worldformer-BART and AWM-BART are compared.

Inputs to the Model			
Observation	**Graph**	**Inventory**	**Valid actions**
[OBS] Location: Connection This area of matted-down crabgrass lies between the vaulted big top entrance to the north and the enticements of the midway to the east, where a sagging banner hangs crookedly above a turnstile. There is a drinking fountain near the side wall of the tent. You can enter the night to the west and south. [OBS] You emerge into the warm night air of summer. Connection [TRANSITION ACT] search balloon	[TRIPLE] you in Connection [TRIPLE] you have balloon [TRIPLE] you have fiberglass pole [TRIPLE] button in Connection [TRIPLE] you have clown mask [TRIPLE] drinking fountain in Connection [TRIPLE] helium in balloon [TRANSITION ACT] search balloon	[OBS] Inventory: You have a clown mask, a fiberglass pole, a balloon and $12.81 to your name.	[ACT] enter [ACT] hit balloon [ACT] close balloon [ACT] breath helium [ACT] look east [ACT] wear mask [ACT] open balloon [ACT] drop pole [ACT] drop mask [ACT] drop balloon [ACT] drop all [ACT] empty balloon [ACT] search balloon
Human Error Annotation			
OLI Errors	**Affordance Errors**	**Insufficient Interaction Coverage Errors**	
take passage, drink bucket,		open balloon, empty balloon, search balloon, hit balloon, drop mask, drop all, drop pole, look east, drop balloon	
Automatic Error Annotation			
OLI Errors	**Affordance Errors**	**Insufficient Interaction Coverage**	
take passage, drink bucket,		open balloon, empty balloon, search balloon, hit balloon, drop mask, drop all, drop pole, look east, drop balloon,	

Table 7: Errors analysis of a validation set example from Ballyhoo: OLI errors occur because there is no passage or bucket in the environment.

Inputs to the Model			
Observation	**Graph**	**Inventory**	**Valid actions**
[OBS] Location: The passage bends from northwest to east and there is a flight of steps down at this point.There is a string of shiny glass beads here. There is a dull toadstone here [OBS] You are holding: A lamp (which is on) [TRANSITION ACT] take all	[TRIPLE] (STONE) in (BEND1) [TRIPLE] (BEADS) in (BEND1) [TRIPLE] you in (BEND1) [TRIPLE] you have (LAMP) [TRANSITION ACT] take all	[OBS] Inventory: You are holding: A lamp (which is on)	[ACT] take inventory [ACT] take toadstone [ACT] take beads [ACT] take all [ACT] examine lamp [ACT] drop lamp [ACT] east [ACT] down [ACT] northwest
Human Error Annotation			
OLI Errors	**Affordance Errors**	**Insufficient Interaction Coverage Errors**	
take all		'eat toadstone', 'drop beads', 'examine toadstone', 'drop toadstone', 'east', 'northwest', 'examine beads', 'down', 'drop all'	
Automatic Error Annotation			
OLI Errors	**Affordance Errors**	**Insufficient Interaction Coverage**	
take all		'eat toadstone', 'drop beads', 'examine toadstone', 'drop toadstone', 'east', 'northwest', 'examine beads', 'down', 'drop all'	

Table 8: Errors analysis of a validation set example from Loose. The action 'take all' will not cause a change in the environment after taking transition action 'take all'.

Inputs to the Model			
Observation	**Graph**	**Inventory**	**Valid actions**
[OBS] Location: Blue Room In the far corner of this tented enclosure a thick, undulating cloud of smoke hovers over a poker game. Straight across from you a tight-jawed dealer stands over a blackjack table covered with a green floor-length tablecloth. The spring-loaded secret panel slides shut. [OBS] Blue Room In the far corner of this tented enclosure a thick, undulating cloud of smoke hovers over a poker game. Straight across from you a tight-jawed dealer stands over a blackjack table covered with a green floor-length tablecloth. The spring-loaded secret panel slides shut. [TRANSITION ACT] wear ribbon	[TRIPLE] you have fiberglass pole [TRIPLE] blackjack table in Blue Room [TRIPLE] blue dot in ticket [TRIPLE] you have ticket [TRIPLE] pink dot in ticket [TRIPLE] you have spreadsheet [TRIPLE] you have bucket [TRIPLE] Comrade Thumb in Blue Room [TRIPLE] uniform in Comrade Thumb [TRIPLE] you have scrap newsprint [TRIPLE] deck cards in Blue Room [TRIPLE] you in Blue Room [TRIPLE] you have skeletkey [TRIPLE] you have ribbon [TRIPLE] you have transistor radio [TRIPLE] dealer in Blue Room [TRIPLE] you have rawhide bullwhip [TRIPLE] radio dial in transistor radio [TRIPLE] cloud smoke in Blue Room [TRANSITION ACT] wear ribbon	[OBS] Inventory: You have your ticket, a bucket, a ribbon, a rawhide bullwhip, a skeleton key, a scrap of newsprint, a fiberglass pole, a transistor radio, a spreadsheet and $13.81 to your name.	[ACT] look dial [ACT] look down comrade thumb [ACT] look at comrade thumb [ACT] close off comrade thumb [ACT] look around dial [ACT] wear ribbon [ACT] drop newsprint [ACT] drop pole [ACT] drop bullwhip [ACT] drop blue [ACT] drop ribbon [ACT] drop spreadsheet [ACT] drop key [ACT] drop bucket [ACT] drop all
Human Error Annotation			
OLI Errors	**Affordance Errors**	**Insufficient Interaction Coverage Errors**	
'drink bucket'		'cut dial with blue', 'cut newsprint with blue', 'put down bullwhip', 'put down spreadsheet', 'put blue in bucket', 'put down newsprint', 'cut bucket with blue', 'cut uniform with blue', 'cut pole with blue', 'put all in bucket', 'put down all', 'break pole with comrade thumb', 'cut spreadsheet with blue', 'cut room with blue', 'put down key', 'put bullwhip in bucket', 'put newsprint in bucket', 'put down pole', 'put down blue', 'put down bucket', 'close off comrade thumb', 'cut comrade thumb with blue', 'cut floor with blue', 'cut bullwhip with blue', 'cut ribbon with blue', 'cut smoke with blue', 'put all around floor', 'put key in bucket', 'cut cards with blue', 'cut key with blue', 'cut dealer with blue', 'look at comrade thumb', 'put spreadsheet in bucket', 'cut poker with blue', 'look around pole'	
Automatic Error Annotation			
OLI Errors	**Affordance Errors**	**Insufficient Interaction Coverage**	
'look dial', 'open cage', 'examin room', 'take ribbon', 'drink bucket'		'cut dial with blue', 'cut newsprint with blue', 'put down bullwhip', 'put down spreadsheet', 'put blue in bucket', 'put down newsprint', 'cut bucket with blue', 'cut uniform with blue', 'cut pole with blue', 'put all in bucket', 'put down all', 'break pole with comrade thumb', 'cut spreadsheet with blue', 'cut room with blue', 'put down key', 'put bullwhip in bucket', 'put newsprint in bucket', 'put down pole', 'put down blue', 'put down bucket', 'close off comrade thumb', 'cut comrade thumb with blue', 'cut floor with blue', 'cut bullwhip with blue', 'cut ribbon with blue', 'cut smoke with blue', 'put all around floor', 'put key in bucket', 'cut cards with blue', 'cut key with blue', 'cut dealer with blue', 'look at comrade thumb', 'put spreadsheet in bucket', 'cut poker with blue', 'look around pole'	

Table 9: Errors analysis of a validation set example from Loose.

Inputs to the Model			
Observation	**Graph**	**Inventory**	**Valid actions**
[OBS] Location: Copier Room The copier room doesn't contain any windows, and vibrates slightly with fluorescent light. A big copier sits quietly in the corner. Doors lead east and west. You can see a Dragon Statue here. [OBS] Copier Room You can see a Dragon Statue here. [TRANSITION ACT] put wire down	[TRIPLE] Copy Machine in Copier Room [TRIPLE] you have Long Key [TRIPLE] Sheet of Paper in Paper Tray [TRIPLE] you have Coil of wire [TRIPLE] you have Gun [TRIPLE] Paper Tray in Copier Room [TRIPLE] Copy Machine Lights in Copier Room [TRIPLE] you have Long Ladder [TRIPLE] Copy Machine Lid in Copier Room [TRIPLE] you in Copier Room [TRIPLE] Dragon Statue in Copier Room [TRIPLE] Copier Room west Stationary Cupboard [TRANSITION ACT] put wire down	[OBS] Inventory: You are carrying: a Long Key a Coil of wire a Long Ladder a Gun	[ACT] take statue [ACT] close paper tray [ACT] west [ACT] put wire down [ACT] put ladder down [ACT] put key down [ACT] put gun down [ACT] put all down [ACT] empty paper tray [ACT] take all off paper tray [ACT] east
Worldformer-BART Predictions			
True Positives	**False Positives**	**False Negatives**	
'take all', 'close paper tray', 'west', 'east'	'take key'	'take wire', 'take statue', 'put ladder down', 'put key down', 'put gun down', 'put all down', 'empty paper tray', 'take all off paper tray'	
AWM-BART Predictions			
True Positives	**False Positives**	**False Negatives**	
'east', 'take all', 'take statue', 'take wire', 'put key down', 'put all down', 'empty paper tray', 'close paper tray', 'west', 'take all off paper tray'		'put ladder down', 'put gun down'	

Table 10: Comparison of model predictions on a test set example from Ludicorp.

Inputs to the Model			
Observation	**Graph**	**Inventory**	**Valid actions**
[OBS] Location: Cave Mouth This is a cave mouth, at one end of a road which winds southeast over rising ground. The entrance west to the caves is a dark tunnel, and only a footpath runs further north, into gorse. The iron door stands open. You can also see a silver coin and a spell book here. [OBS] You close the cedarwood box. [TRANSITION ACT] close door	[TRIPLE] Helistar's grimoire in cedarwood box [TRIPLE] you have cedarwood box [TRIPLE] you have magic burin [TRIPLE] you have cube [TRIPLE] spell book in Cave Mouth [TRIPLE] you in Cave Mouth [TRIPLE] players coin in Cave Mouth [TRIPLE] you have beautiful red carpet [TRIPLE] iron door in Cave Mouth [TRANSITION ACT] close door	[OBS] Inventory: You are carrying: a beautiful red carpet the "chasm" cube the "cave" cube a cedarwood box (which is closed) a magic burin	[ACT] west [ACT] take book [ACT] take coin [ACT] take all [ACT] close door [ACT] put box down [ACT] put carpet down [ACT] put burin down [ACT] put cave down [ACT] put all down [ACT] examine book [ACT] open box [ACT] north [ACT] up
Worldformer-BART Predictions			
True Positives	**False Positives**	**False Negatives**	
'take book', 'open box', 'up', 'north', 'take coin'	'take sword', 'push sword to marble', 'throw lantern at sword'	'take all', 'put box down', 'put carpet down', 'put burin down', 'put cave down', 'put all down', 'examine book', 'open door'	
AWM-BART Predictions			
True Positives	**False Positives**	**False Negatives**	
'take book', 'put all down', 'open box', 'up', 'examine book', 'open door', 'north', 'take all', 'take coin', 'put box down'	'examine door'	'put carpet down', 'put burin down', 'put cave down'	

Table 11: Comparison of model predictions on a test set example from Balances.

Inputs to the Model			
Observation	**Graph**	**Inventory**	**Valid actions**
[OBS] Location: Torch Room This is a large room with a prominent doorway leading to a down staircase. Above you is a large dome. Up around the edge of the dome (20 feet up) is a wooden railing. In the center of the room sits a white marble pedestal. A piece of rope descends from the railing above, ending some five feet above your head. Sitting on the pedestal is a flaming torch, made of ivory. [OBS] Torch Room This is a large room with a prominent doorway leading to a down staircase. Above you is a large dome. Up around the edge of the dome (20 feet up) is a wooden railing. In the center of the room sits a white marble pedestal. A piece of rope descends from the railing above, ending some five feet above your head. Sitting on the pedestal is a flaming torch, made of ivory. [TRANSITION ACT] put down knife	[TRIPLE] you have brass lantern [TRIPLE] pedestal in Torch [TRIPLE] you have clove garlic [TRIPLE] you have nasty knife [TRIPLE] you have sword [TRIPLE] torch in pedestal [TRIPLE] you in Torch [TRANSITION ACT] put down knife	[OBS] Inventory: You are carrying: A sword A nasty knife A brass lantern (providing light) A clove of garlic	[ACT] south [ACT] put out lantern [ACT] take torch [ACT] put down lantern [ACT] put down knife [ACT] put down garlic [ACT] put down sword [ACT] put down all [ACT] put knife on marble [ACT] put garlic on marble [ACT] throw lantern at knife [ACT] put all on marble
Worldformer-BART Predictions			
True Positives	**False Positives**	**False Negatives**	
'put out lantern', 'take torch', 'south'	'take sword', 'push sword to marble', 'throw lantern at sword'	'take knife', 'take all', 'put down lantern', 'put down garlic', 'put down sword', 'put down all', 'put garlic on marble', 'throw lantern at knife'	
AWM-BART Predictions			
True Positives	**False Positives**	**False Negatives**	
'put out lantern', 'throw lantern at knife', 'take knife', 'south', 'take torch', 'put garlic on marble', 'put down all', 'put down lantern', 'take all'	'put knife on marble'	'put down garlic', 'put down sword'	

Table 12: Comparison of model predictions on a test set example from Zork1.

Inputs to the Model			
Observation	**Graph**	**Inventory**	**Valid actions**
[OBS] Location: West Royal Road This road is quite beautiful, decorated on its sides with fluorescent mosses that feed on the minerals in the stones that line the sides of the roads. Somehow, the mosses do not leave their designated stones. High walls on both sides make the street feel more like a hall than an open passageway, and gates leading to palaces break up the monotony of the stone. A single gate is open to the north. The road continues east and to the west is the outer court of the Lord's Palace. [OBS] West Royal Road This road is quite beautiful, decorated on its sides with fluorescent mosses that feed on the minerals in the stones that line the sides of the roads. Somehow, the mosses do not leave their designated stones. High walls on both sides make the street feel more like a hall than an open passageway, and gates leading to palaces break up the monotony of the stone. A single gate is open to the north. The road continues east and to the west is the outer court of the Lord's Palace. [TRANSITION ACT] put on shield	[TRIPLE] you have pickaxe [TRIPLE] moss in West Royal Road [TRIPLE] you have King's Order [TRIPLE] you have gear [TRIPLE] you have green moss [TRIPLE] Kraxis in West Royal Road [TRIPLE] you have Sword [TRIPLE] ground in West Royal Road [TRIPLE] you have shield [TRIPLE] you in West Royal Road [TRIPLE] you have lantern [TRIPLE] you have magical torch [TRANSITION ACT] put on shield	[OBS] Inventory: You are carrying: a green moss (providing light) a gear a pickaxe a magical torch (providing light) a Sword a shield King's Order a lantern (providing light)	[ACT] say manaz [ACT] north [ACT] west [ACT] put order down [ACT] put moss down [ACT] put lantern down [ACT] put pickaxe down [ACT] put shield down [ACT] put torch down [ACT] put sword down [ACT] put gear down [ACT] put all down [ACT] put on shield [ACT] wield sword [ACT] east
Worldformer-BART Predictions			
True Positives	**False Positives**	**False Negatives**	
'west', 'east', 'put all down', 'north', 'say manaz'	'take shield'	'take off shield', 'put order down', 'put moss down', 'put lantern down', 'put pickaxe down', 'put shield down', 'put torch down', 'put sword down', 'put gear down', 'wield sword'	
AWM-BART Predictions			
True Positives	**False Positives**	**False Negatives**	
'north', 'west', 'east', 'put torch down', 'put pickaxe down', 'take off shield', 'wield sword', 'say manaz'	'examine door'	'put order down', 'put moss down', 'put lantern down', 'put shield down', 'put sword down', 'put gear down','put all down'	

Table 13: Comparison of model predictions on a test set example from Deephome.

<table>
<tr><th colspan="4">Inputs to the Model</th></tr>
<tr><th>Observation</th><th>Graph</th><th>Inventory</th><th>Valid actions</th></tr>
<tr><td>[OBS] Location: Storage
This is a minor storage chamber, connected to the study through a doorway in the northern wall. There are some shelves, mostly vacant, here.
On one of the shelves are four small vials, each neatly labelled and containing some green powder. [OBS] Storage
This is a minor storage chamber, connected to the study through a doorway in the northern wall. There are some shelves, mostly vacant, here.
On one of the shelves are four small vials, each neatly labelled and containing some green powder.
[TRANSITION ACT] take all off shelves</td><td>[TRIPLE] mysterious vial in shelves [TRIPLE] you in Storage [TRIPLE] shelves in Storage [TRIPLE] dark tower in Storage [TRIPLE] Storage south Study [TRANSITION ACT] take all off shelves</td><td>[OBS] Inventory: You are carrying nothing.</td><td>[ACT] take vial [ACT] take all off shelves [ACT] north</td></tr>
<tr><th colspan="3">Worldformer-BART Predictions</th></tr>
<tr><th>True Positives</th><th>False Positives</th><th>False Negatives</th></tr>
<tr><td>’north’</td><td>’take vial’</td><td>’put vial down’, ’put all down’, ’put vial on shelves’, ’put all on shelves’</td></tr>
<tr><th colspan="3">AWM-BART Predictions</th></tr>
<tr><th>True Positives</th><th>False Positives</th><th>False Negatives</th></tr>
<tr><td>’put vial down’, ’put vial on shelves’, ’put all on shelves’, ’north’</td><td></td><td>’put all down’</td></tr>
</table>

Table 14: Comparison of model predictions on a test set example from Temple.