International Journal of User-System Interaction 13(1) 2020, 18-31

Interactive language learning - How to explore complex environments using natural language?

Tatiana-Andreea Petrache[1], Traian Rebedea[1], Ştefan Trăuşan-Matu[1,2,3]

[1]Universitatea Politehnica din Bucureşti
Splaiul Independenţei nr. 313, Bucureşti
E-mail: tatianap19@gmail.com, traian.rebedea@upb.ro, stefan.trausan@upb.ro
[2]Institutul de Cercetări în Inteligenţa Artificială
Calea 13 Septembrie nr. 13, Bucureşti
[3]Academia Oamenilor de Ştiinţă din România
Splaiul Independenţei nr. 54, Bucureşti

Abstract. Implicit knowledge about the physical world we live in is gained almost effortlessly through interaction with the environment. In the same manner, this knowledge cannot be simply inferred from language, as humans normally avoid stating what is trivially implied or observed in the world. This paper is about a novel perspective into progressing artificial intelligence toward understanding the true language meaning through interaction with complex environments. The arising field of text-based games seems to hold the key for such an endeavour. Text-based games placed in a reinforcement learning formalism have the potential of being a strategic path into advancing real-world natural language applications - the human world itself is one of partial understanding through communication and acting on the world using language. We present a comparative study highlighting the importance of having a unified approach in the area of learning agents to play families of text-based games, with the scope of establishing a benchmark that will enable the community to advance the state of the art. To this end, we will look at the corpora and the first two winner solutions from the competition launched by Microsoft Research - FirstTextWorld Problems. The games from the proposed corpora share the same objective, cooking a meal after collecting ingredients from a modern house environment, having the layout and the recipes change from one game to another.

Keywords: reinforcement learning, natural language processing, text-based games, partial understanding, language meaning

DOI: 10.37789/ijusi.2020.13.1.2

1. Introduction

Reinforcement Learning (RL) is a general-purpose framework for decision

making that has gained a lot of popularity in recent years. The central idea of RL is that, through interactions with a complex environment, the agent receives a set of positive and negative rewards. Its objective is to build an agent capable not only of acting, but also of taking decisions that in the long run will maximize the accumulated rewards. This idea was inspired by behaviorism, summarized by Skinner in his reinforcement theory as "an individual's behavior is a function of its consequences" (Skinner, 2014). But Reinforcement Learning has many "faces", expanding to multiple domains such as: neuroscience (the dopamine system), economics (the Rationality Game Theory), or mathematics (the Optimal Control Theory). All these fields are studying the same problem: optimizing decision making to obtain the best results while exploring a complex environment. This is one of the reasons why Deep Reinforcement Learning - a combination of Deep Learning (DL) and RL for creating efficient algorithms - is considered the pathway for achieving general Artificial Intelligence (AI), as Silver (2019) simply puts it: "AI = DL + RL".

Researchers have been successfully applying reinforcement learning for various applications with a reduced action space, such as surpassing human performance on the Atari benchmark (Badia, 2020). This paper is focusing on reinforcement learning in Natural Language Processing (NLP), a task less approached due to the combinatorial and compositional nature of the problem, that makes it difficult to optimize. With the popularity of text-based games, solving this task has become more approachable and its successful resolution will have a significant contribution on learning to behave in dialogue like environments (Jurafsky, 2014).

A profound language comprehension requires language to be rooted in the environment itself and goes beyond language modelling, as described by Forbes et al. (2019). There is an intrinsic limitation of how much a system can learn from language, as words often refer to actions and objects outside the language. Similarly, Wittgenstein stated that the meaning of a word has to be found in its use, via the so-called model of "language-games" (Wittgenstein, 2009).

Text-based games or Interactive Fiction (Montfort, 2005) are structured as a conversation or "dialogue" between the game engine and the user and they represent a modern interpretation of Wittgenstein's theory of meaning (Wittgenstein, 2009). The game engine generates a textual description of the game state and the player inputs a command in natural language to execute

an action (see Figure 1). Once the command is interpreted by the parser and executed, a new game state is returned, followed by another player's command. This process repeats until the game is completed.

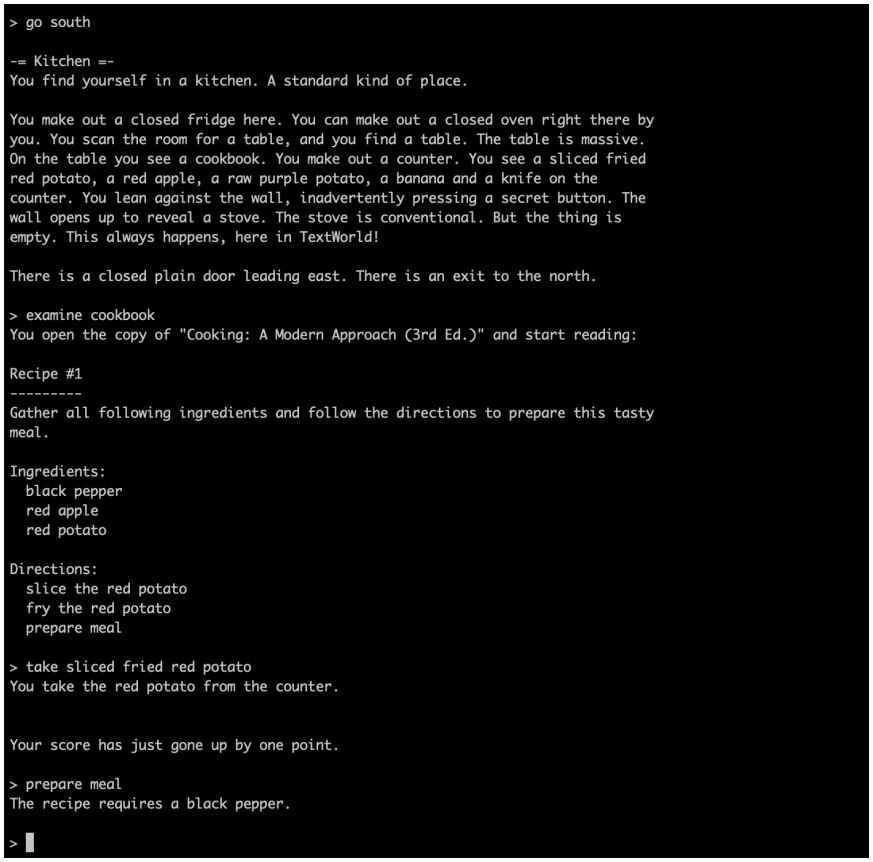

Figure 1: Interaction example from a cooking text-based game generated using the TextWorld framework: lines marked with ">" represent user's input followed by the game engine feedback (output obtained from personal experimental setup)

Generally, the game state is partially observable, as the agent has no full access to the underlying environment. Another challenging aspect is given by the innate ambiguity of textual descriptions (in the example above the player has to carefully note that the red potato is already fried and sliced). The action space differs from a state to another and can consist of a (usually short) sequence of words from a large vocabulary, thus making it potentially

unlimited. Furthermore, in most games the rewards are sparse.

In this textual setup, the player has the challenging task of finding an optimal policy that will maximize the total reward. Similarly, to progress in such games, an AI agent has to generate contextually relevant commands and possess other skills, such as: long-term memory, exploration, and common-sense reasoning. Consequently, most human-made text games are beyond the capabilities of the current NLP learning algorithms. To gain an intuition regarding how vast is the action space from these text games, let's consider one of the most popular text-based games, Zork[7], where a player has to come up with commands of up to five-words from a vocabulary of 697 words allowed by the game's parser. This vocabulary leads to a total of up to 697^5 possible commands for the learning agent to choose from at every interaction with the game engine.

Considering the increasing popularity of RL-based agents for text-based games, in this paper we will present a comparative study with the scope of establishing a benchmark that will enable the community to advance the state of the art in this area.

The paper continues as follows. In Section 2 we will describe the prior related work along with some important directions of research. In Section 3 we will provide a summary with the results obtained from replicating the code provided by the first and second place solutions in the FirstTextWorld Problems competition. In Section 4 we will highlight the main findings gained from the experiments.

2. Related research for RL-based agents to play text-based games

In this section we look at prior work from different perspectives: widely cited text-based game playing agents, methods for reducing the exponential action space, techniques to allow knowledge to be transferred to unseen games and environments, employed reinforcement learning algorithms and frameworks.

---

[7] http://www.infocom-if.org/games/zork1/zork1specs.html

2.1. Seminal works

Due to the challenges outlined in the previous section, the domain of text-based games is modestly covered in AI research. There are two seminal papers being widely cited, which propose different neural architectures for solving text-based games.

Narasimhan et al. (2015) proposed an LSTM-DQN architecture that consists of two parts: an LSTM network (Hochreiter and Schmidhuber, 1997) to represent the textual input and a DQN (Deep Q-Network) (Mnih et al., 2015) that returns Q-values for the state and action representations. The authors choose to simplify the action space to two-word pairs in the form <*verb-object*> and provide the resulting set as available commands. The two Q-values outputted by the DQN model are averaged to get one final score. The authors test their method on two games ("Homeworld" and "Fantasyworld") using the Evennia[8] framework. A limitation of the LSTM-DQN architecture is related to the fact that the two action Q-values are finally averaged, which can be a problematic aspect when verbs and objects are not independent. For example, the value of the verb "drink" depends on the object; consider the difference between the values of "drink" when followed by either "water" or "poison" objects.

He et al. (2015) introduced a Deep Reinforcement Relevance Network (DRRN) for playing text-based games. DRRN uses separate embeddings (bag-of-words representations) for states and actions and then applies feedforward networks with a variable number of hidden layers to obtain Q-values. The final Q-value is the inner product of the state and action Q-values. The authors evaluated the DRRN, separately, on two games ("Saving John" and "Machine of Death"), different from the ones used for evaluating the aforementioned LSTM-DQN agent. The main drawback of DRNN is the use of bag-of-words for input representation, meaning that the model is incapable of differentiating simple nuances in the state, such as: "There is a treasure chest to your left and a dragon to your right." and "There is a treasure chest to your right and a dragon to your left."

[8] http://www.evennia.com

2.2. Unbounded Action Space

The potential action space in text-based games is large, dictated by the innate combinatorial nature of the problem. For example, the action space in Zork can reach up to 697^5 of possible commands. Due to this aspect, we define the action space in text-based games as exponential for current text-games, but practically unbounded in real-world scenarios.

In order to solve the problem of unbounded action space, several ideas for reducing the size of the action space in text games were recently proposed.

In a first attempt, Fulda et al. (2017) make use of embeddings to deduce object affordances. Later, Zahavy et al. (2018) reduce the vast action space by making use of a playthrough of the game to store important features in a replay buffer and they propose an action elimination network (AEN) that is trained along with the Deep Q-Network, learning to remove actions that are implausible to change the game state by relying on the game engine for feedback. Recently, Hausknecht et al. (2020) proposed the adoption of action generation based on templates provided by the TextWorld framework, as a way to limit the action space.

2.3. Generalization and transfer learning

A successful agent needs to have generalization capabilities to be able to transfer its learned skills to never-before-seen games. Therefore, transfer learning capabilities could be augmented by employing external knowledge to the learning agents.

In a similar manner, Hausknecht et al. (2019) demonstrate that pretrained language models along with the addition of heuristics for sub-policies can significantly improve the ability of an agent to solve text-based games.

Also, previous research has shown that many NLP tasks can be formulated as instances of question-answering (QA) and that we can transfer knowledge between these tasks (McCann et al., 2017). Similarly, Ammanabrolu and Riedl (2019) demonstrate how pre-training certain parts of their Knowledge Graph DQN (KG-DQN) using existing question-answering methods improves convergence and allows knowledge to be transferred across different text-based games.

2.4. Reinforcement Learning algorithms

Since the release of LSTM-DQN, a variety of researchers have used some form of Deep Q-Network to solve text-based games. However, Adolphs et al. (2019) showed that scoring commands using an advantage-actor-critic method (Mnih et al., 2016) offers a significant improvement in performance and convergence over the previous Deep Q-Network variants, while encoding the input states using bi-directional Gated Recurrent Units (GRUs) (Cho et al., 2014).

2.5. Frameworks

To enable agents to progress towards navigating text-based games in an accessible and controllable manner, a series of frameworks were proposed by the research community that ease agent development and assessment.

One of the most popular frameworks is TextWorld (Côté et al., 2018), a sandbox for generating text-based games starting from predefined configurations for the game layout, difficulty or the type of reward (dense, balanced or sparse). TextWorld provides a simple API that facilitates a learning agent to interact with the game engine similarly to OpenAI's gym (Brockman, 2016), making it a perfect environment to nurture the application of RL algorithms.

Yuan et al. (2019) address the notion of a novel interactive question-answering task, dubbed QAi, with the scope of modelling question-answering tasks using TextWorld. Urbanek et al. (2019) introduce Light, a dataset of text-based games constructed with crowdsourcing efforts to enable multiple agents to learn how to generate meaningful conversations and emotional reactions.

Hausknecht et al. (2019) have open-sourced Jericho, an optimized framework for playing human-constructed text-based games - as a way of adapting real textual games to the RL formalism. In addition to providing action templates and vocabulary that can reduce the size of the action space, Jericho comes with other features to make text-based games more accessible to existing agents, such as: tree-like representation of the objects encountered in the one's world or state-change detection and valid-action identification.

3. A new unified approach for learning agents to play text-based games

The outlined summary about related research on agents playing text-based games shows a lack of common ground when it comes to having a reproducible benchmark to help the community track progress and moving forward the state of the art. The authors base their results on different game inputs generated by different game simulators, making comparison across this research space quite difficult, if not irrelevant. For this reason, we decided to focus our attention on frameworks and game corpora that empower easy replication and scientific progress.

To this end, we will look at the dataset and solutions proposed for the competition launched by Microsoft Research during NeurIPS 2018 - FirstTextWorld Problems (FTWP): A Reinforcement and Language Learning Challenge. In this competition (which ran between January-July 2019), an agent is placed within a house and is charged with the task of collecting ingredients with the scope of cooking and then eating a meal. At the evaluation phase, the agent has to prove its skills in a new, but similarly configured house.

The dataset[9] is available on the FirstTextWorld Problems competition site. It contains 4440 different training games, 222 validation games, 514 test games. The games in the training set have various difficulty levels dictated by the "skills" that the agent has to acquire in order to succeed in its quests. These skills depend on the number of ingredients in the recipe, the actions to take or the number of locations to visit in the environment.

In this section we will present and analyze the results we noted from replicating the code provided by the first and second place solutions in the FirstTextWorld Problems competition. It is interesting to add that the two solutions were based on different approaches and came up with different formulations for the given problem, many of the ideas employed being already mentioned in the related work.

---

[9] https://competitions.codalab.org/competitions/21557#learn_the_details-data

3.1. First Place - CogniTextWorld

The winning agent, CogniTextWorld is based on a combination of specialized BERT models (Devlin et al., 2018) and employs the UCB1 (optimism in face of uncertainty) algorithm (Kuleshov, 2014) to balance exploration and exploitation. The agent builds a representation of the game state from the feedback provided by two commands: "look" and "inventory". In order to limit the unbounded action space associated with text-based games, the agent took advantage of the requested command templates and further enhanced the empty slots (e.g.: *chop {f} with {o}*) from the template with entities taken from the given input state description, which were learned via a specialized BERT model with a token classification head to perform named-entity recognition.

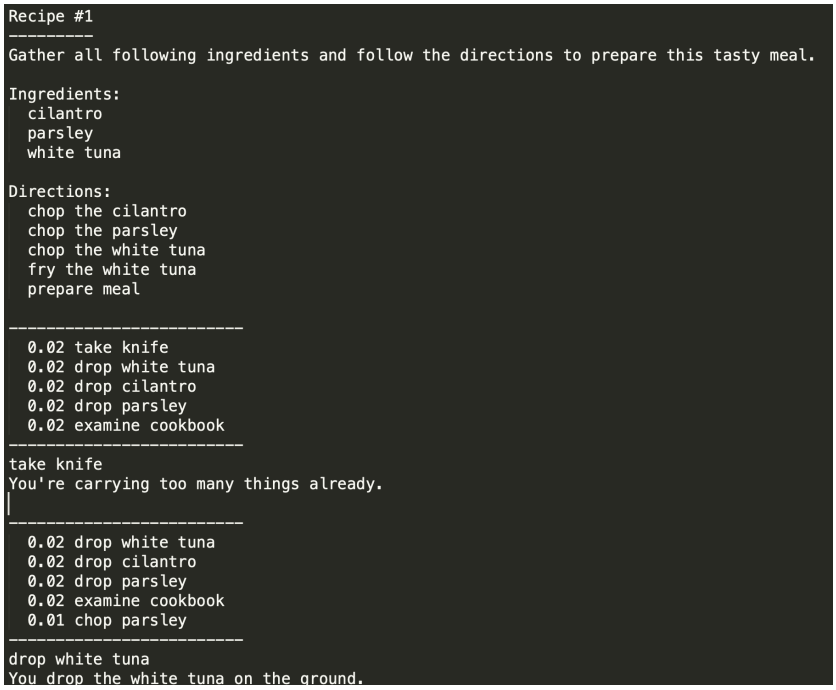

Figure 2: CogniTextWorld's interactions (output obtained from personal experimental setup)

The resulting task, consisting of pairs of game state and constructed potential commands, can be seen as a basic formulation of a question-answering (QA) model which will output probabilities for the potential actions (see Figure 2). This classification task - mapping game states to

commands - was trained using a pre-trained BERT model with a specialized output head.

The agent's final decision relies not only on the probability output from the classification task, but also on the state trajectory history, which enables the agent to deal with the partial observability of the environment in which it has to act. For this purpose, the CogniTextWorld agent makes use of the UCB1 algorithm, maintaining a count of how many times each command has been executed in each state and adjusting the output probabilities from the QA model accordingly to the UCB1 output: it increases the probability assigned to commands used less in the past.

3.2. Second Place - LeDeepChef

The second place agent - LeDeepChef - uses a dual approach for solving the cooking game: the first model is in charge with generating a set of reasonable commands given the input state at any step and the second – representing the actual agent - is trained to rank the commands based on their expected future reward using an advantage-actor-critic algorithm (Mnih et al., 2016).

The authors built a model based on the textual state (represented by a series of inputs such as: observation, location, description, previous commands, unnecessary items) and the list of potential commands that is capable of scoring the current game state (the "critic") and ranking the list of commands (the "actor"). To compensate for the combinatorial action space, the authors use an approach inspired by feudal Reinforcement Learning (Dayan and Hinton, 1993) and managed to reduce the size of the action space by combining multiple actions into "high-level" actions, such as: "take required items from here" or "drop unnecessary items" (see Figure 3).

Moreover, the authors made the learning agent more robust to never-before-seen recipes and ingredients by training with data from external food databases and by using pre-trained GloVe (Pennington et al., 2014) embeddings to project related food items close to each other.

An important problem in Reinforcement Learning is related to sparse rewards, which the authors try to overcome by using an online actor-critic algorithm that computes the reward at specific steps within an episode.

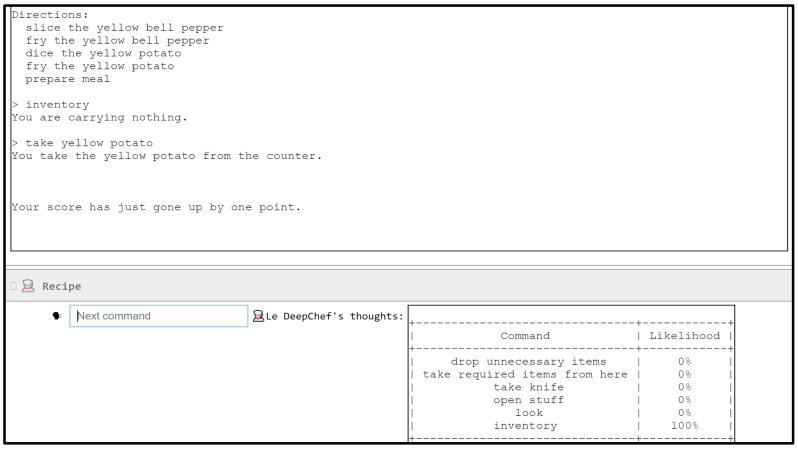

Figure 3: LeDeepChef's interactions (output obtained from personal experimental setup)

3.3. Experimental results

As mentioned, one of the main objectives of the paper was to assess the performance of different RL agents proposed for text-based games. The main personal contribution consists in explaining the results and providing clear directions on how to advance the progress in this area based on these results. Using a personal experimental setup, we have tested several agents as shown in Table 1. The scores obtained from the test set show that CogniTextWorld and LeDeepChef achieve significant better results when compared to other related baseline methods, such as LSTM-DQN (Narasimhan et al., 2015) and DRRN (He et al., 2016).

The standard baselines (LSTM-DQN and DRNN) are not able to surpass a test score of 14%, indicating that they are not suitable to be applied "as is" on the specific task of solving "never-before-seen" text-based games. The significant difference in score between DRNN and LSTM-DQN is mainly related to the larger action-space (based on pre-selected two-word pair commands) that the LSTM-DQN method has to deal with as opposed to ranking only the available commands in the case of DRRN. As an improvement, the list of valid actions provided by the TextWorld framework could be used when training the LSTM-DQN method as a supervised loss to guide the agent toward actions that will result in state changes, thus facilitating a faster convergence to the optimal policy.

Table 1: Experimental comparison of different RL agents for the FirstTextWorld competition

Method	Test Score (Penalized)[10]
LSTM-DQN	1%
DRNN[11]	13.2%
LeDeepChef	69.3%
CogniTextWorld	70.8%

The two other learning agents, LeDeepChef and CogniTextWorld, prove significantly better generalization results across the new 514 text-based games from the test set, mostly due to their approach to reduce the action space through the use of command templates and powerful action generation.

The fact that the CogniTextWorld agent outperforms LeDeepChef serves to highlight the advantage of representations from powerful, pre-trained language models (such as BERT) to build effective and general policies. It becomes also clear that under the right constraints and by populating command templates with learned entities, the problem of large, compositional action spaces can be tackled successfully.

On the other hand, these results demonstrate that the games provided in the competition do not include quests needing complex long-term planning, meaning that there is still a significant gap between the generated TextWorld games from the competition and the hand-authored text-based games. This realization opens up new areas of research, such as exploring the open-sourced Jericho framework (Hausknecht et al., 2019) to start learning an agent to play real text-based games. Similarly, the experiments presented in this section point out that transformer-based neural architectures are capable of yielding impressive gains even for text-based tasks. We expect these advances may be applicable to human-made text-based games from Jericho, but will need to be adapted from a supervised training regime into reinforcement learning.

4. Conclusion

This paper focused on the underexplored research field of solving families of

[10] Score with handicap penalty

[11] Test score result for the DRNN agent was taken from Adolphs et al. (2019)

text-based games, with all the associated challenges, such as: deeper language understanding and use, but also memory, planning, and exploration - all active areas of research in Artificial Intelligence. Moreover, we presented two agents improving upon standard baseline methods.

It is important to take away the formulation of this task in the form of a question-answering model along with a smart command generation technique, which proved to be a successful recipe for building effective, general policies and opens the promise for new exciting achievements.

The adventure is not over here, as text-adventure games represent a steppingstone toward deeper language understanding, a relevant mirror for the intricacies of the human world, which is one of partial understanding through interaction with the world using language.

References

Adolphs, L., & Hofmann, T. (2019). Ledeepchef: Deep reinforcement learning agent for families of text-based games. arXiv preprint arXiv:1909.01646.

Ammanabrolu, P., & Riedl, M. O. (2019). Transfer in deep reinforcement learning using knowledge graphs. arXiv preprint arXiv:1908.06556.

Badia, A. P., Piot, B., Kapturowski, S., Sprechmann, P., Vitvitskyi, A., Guo, D., & Blundell, C. (2020). Agent57: Outperforming the atari human benchmark. arXiv preprint arXiv:2003.13350.

Brockman, G., Cheung, V., Pettersson, L., Schneider, J., Schulman, J., Tang, J., & Zaremba, W. (2016). Openai gym. *arXiv preprint arXiv:1606.01540*.

Cho, K., Van Merriënboer, B., Bahdanau, D., & Bengio, Y. (2014). On the properties of neural machine translation: Encoder-decoder approaches. *arXiv preprint arXiv:1409.1259*.

Côté, M. A., Kádár, Á., Yuan, X., Kybartas, B., Barnes, T., Fine, E., Tay, W. (2018, July). Textworld: A learning environment for text-based games. In Workshop on Computer Games (pp. 41-75). Springer, Cham.

Dayan, P., & Hinton, G. E. (1993). Feudal reinforcement learning. In Advances in neural information processing systems (pp. 271-278).

Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Forbes, M., Holtzman, A., & Choi, Y. (2019). Do Neural Language Representations Learn Physical Commonsense?. arXiv preprint arXiv:1908.02899.

Fulda, N., Ricks, D., Murdoch, B., & Wingate, D. (2017). What can you do with a rock? affordance extraction via word embeddings. arXiv preprint arXiv:1703.03429.

Hausknecht, M. J., Ammanabrolu, P., Côté, M. A., & Yuan, X. (2020). Interactive Fiction Games: A Colossal Adventure. In AAAI (pp. 7903-7910).

Hausknecht, M., Loynd, R., Yang, G., Swaminathan, A., & Williams, J. D. (2019). Nail: A general interactive fiction agent. arXiv preprint arXiv:1902.04259.

He, J., Chen, J., He, X., Gao, J., Li, L., Deng, L., & Ostendorf, M. (2015). Deep reinforcement learning with a natural language action space. arXiv preprint arXiv:1511.04636.

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, *9*(8), 1735-1780.

Jurafsky, D., & Martin, J. H. (2014). Speech and language processing. Vol. 3.

Kuleshov, V., & Precup, D. (2014). Algorithms for multi-armed bandit problems. arXiv preprint arXiv:1402.6028

McCann, B., Keskar, N. S., Xiong, C., & Socher, R. (2018). The natural language decathlon: Multitask learning as question answering. arXiv preprint arXiv:1806.08730.

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

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., ... & Petersen, S. (2015). Human-level control through deep reinforcement learning. *nature*, *518*(7540), 529-533.

Montfort, N. (2005). Twisty Little Passages: an approach to interactive fiction. Mit Press.

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

Pennington, J., Socher, R., & Manning, C. D. (2014, October). Glove: Global vectors for word representation. In Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP) (pp. 1532-1543).

Silver, D. (2020, January 19). David Silver on Deep Learning + RL = AI?. Retrieved from https://www.coursera.org/lecture/prediction-control-function-approximation/david-silver-on-deep-learning-rl-ai-xZuSl

Skinner, B. F. (2014). Contingencies of reinforcement: A theoretical analysis (Vol. 3). BF Skinner Foundation.

Urbanek, J., Fan, A., Karamcheti, S., Jain, S., Humeau, S., Dinan, E., Weston, J. (2019). Learning to speak and act in a fantasy text adventure game. arXiv preprint arXiv:1903.03094.

Wittgenstein, L. (2009). Philosophical investigations. John Wiley & Sons.

Yuan, X., Côté, M. A., Fu, J., Lin, Z., Pal, C., Bengio, Y., & Trischler, A. (2019). Interactive language learning by question answering. arXiv preprint arXiv:1908.10909.

Zahavy, T., Haroush, M., Merlis, N., Mankowitz, D. J., & Mannor, S. (2018). Learn what not to learn: Action elimination with deep reinforcement learning. In Advances in Neural Information Processing Systems (pp. 3562-3573).