

Dungeons and DQNs: Toward Reinforcement Learning Agents that Play Tabletop Roleplaying Games

Lara J. Martin
ljmartin@gatech.edu

Srijan Sood
srijansood@gatech.edu

Mark O. Riedl
riedl@cc.gatech.edu

School of Interactive Computing, Georgia Institute of Technology
Atlanta, GA 30332

Abstract

Game playing has been an important testbed for artificial intelligence. Board games, first-person shooters, and real-time strategy games have well-defined win conditions and rely on strong feedback from a simulated environment. Text adventures require natural language understanding to progress through the game but still have an underlying simulated environment. In this paper, we propose tabletop roleplaying games as a challenge due to an infinite action space, multiple (collaborative) players and models of the world, and no explicit reward signal. We present an approach for reinforcement learning agents that can play tabletop roleplaying games.

1 Introduction

Computer games have long been used as a testbed for measuring progress in artificial intelligence. Computer games provide complex, dynamic environments that are more complicated than made-up toy problems but less complicated than the real world [LvL01]. Artificial intelligence systems have been demonstrated to play board games, Atari games, first-person shooters, and multiplayer online battle arena games at or above human-level performance [SHM⁺16, MKS⁺13, LC16, Ope]. These types of games have large, fixed sets of actions and a large number of possible, non-ambiguous states, though the environment may only be partially observable. They also have well-defined win conditions and/or scores that can be used by an agent to determine if it is playing the game well. Some progress has also been made in playing text adventure games, or Interactive Fiction (IF) [NKB15, KKKR17, HZMM18, YCS⁺18]. In IF, the agent must infer the true, underlying state of the world from natural language descriptions then choose from a large but fixed set of actions. While text adventure games capture the ambiguous nature of some real world tasks, they are structured as puzzle games with an underlying game engine that maintains a single, ground-truth state and dictates what actions are legal or not.

In tabletop roleplaying games (TRPGs), a group of players construct artificial personas for themselves and describe the actions that their personas take within a shared, imaginary world. One of the most popular variants is Dungeons & Dragons (D&D) [GA74]. It and other similar variants assign one player to a special role called the Game Master (GM)—sometimes called the Dungeon Master (DM)—whose job is to act as arbiter, enforcing an agreed-upon set of rules pertaining to the more formulaic parts of roleplaying, such as combat, and to dictate the actions of any additional characters called non-player characters (NPCs). In this paper, we will discuss the challenges of creating an intelligent agent capable of playing D&D.

Since TRPGs have traditionally been discourse-based, the space of actions is infinite, constrained only in a few circumstances by rules. This is unlike IF where there are usually only a predefined set of actions, and

Copyright © by the paper's authors. Copying permitted for private and academic purposes.

In: A. Editor, B. Coeditor (eds.): Proceedings of the XYZ Workshop, Location, Country, DD-MMM-YYYY, published at <http://ceur-ws.org>

those actions serve the purpose of solving puzzles to unlock the story. In TRPGs, even though the players can choose actions, none of the players know exactly what will happen in response to those actions and should adjust accordingly. Even the Game Master may encounter valid player actions that are unexpected, and they must decide how aspects of the world that are not controlled by the players will respond. Most significantly, no single player or system—including the Game Master—possesses a ground-truth understanding of the complete state of the world.

In a game like D&D, actions cannot be cleanly mapped to states. Instead, players need to maintain a general model of the world that can be flexibly altered as the story progresses. Since there is no shared simulation engine that maintains a ground-truth state of the world, there is no way for players to receive feedback about the consequences of their actions except for intrinsic motivation. This means that an AI player would need a set of commonsense knowledge and procedures so that it can act in a reasonable manner. The AI should know what can physically and temporally happen in the world (e.g. if I leave the lightsaber here, it will stay here until someone picks it up again); what social and cultural norms it should follow (e.g. greet people when you meet them); and what tropes the genre normally follows (e.g. fairies are found in forests).

Action selection in TRPGs can be further complicated by the fact that there is no well-defined win condition. TRPGs are usually set up with scenarios called *campaigns* where there are short-term objectives (such as quests) to complete, but even those might not be clearly defined. In D&D, characters may die, and “hit points” (numerical indication of health) can be thought of as an indicator of success in combat, but there are no clear signals of success or progress in non-combat portions (the majority) of the game. This makes it especially hard for an AI player to know whether it is acting appropriately (i.e. there is no explicit reward signal).

D&D is also largely collaborative, which is unusual for a game with multiple players. Collaboration in a game means that not only does the agent need to understand what their fellow players are trying to do but be able to work toward a joint goal which might not be explicit. The agent should not be just fulfilling its own agenda.

In this paper, we propose an approach to creating a TRPG player. Since this is an expansive challenge for the current state of AI, we will focus on the improvisational nature of action selection in the context of a quest. We have made the following simplifying assumptions in order to initially make the challenge more tractable. (1) We do not consider combat or actions that are constrained by numerical values such as strength or health. (2) We also assume that the agent is always “in character” and thus does not interact with other players in extra-dietetic ways (e.g., out of character conversations to plan out actions). (3) If another player is a GM, we only consider descriptions of events that occur, but not refereeing communications. The important aspects that we’re still maintaining are collaboration, improvisation, and keeping track of and maintaining a consistent world. The world is represented as a set of rules acting on the current state, informed by a sense of genre.

In the remainder of the paper, we relate TRPG playing to interactive fiction, interactive storytelling, and story generation. We put forth a proposal for using a form of reinforcement learning—Deep Q Networks (DQNs)—to meet the criteria above for the portions of TRPGs we focus on.

2 Background and Related Work

Interactive Fiction (IF) has been around long before the creation of the personal computer, found in the form of choose-your-own-adventure books. These stories enabled the user to not only experience the narrative but to have input into what events will take place. Computerized IF provided users with flexibility of input; giving the player natural-language commands provided a greater sense of agency. Early computer IF, such as *Adventure* and *Zork*, had a command language in the form of ⟨verb⟩ ⟨noun phrase (NP)⟩, which can include a prepositional phrase and/or adjectives. For example, “enter house” or “give book to woman”. Patterns like this can be resolved by simple grammars, but the language that these systems permitted was very limited. Early IF-playing systems avoided dealing with language all together, having the system work only with propositional logic for example [HA04]. Recent work has focused on neural networks and reinforcement learning—in particular deep Q networks (DQNs)—to play IF [NKB15, HZMM18, YCS⁺18]. IF has become an increasingly common testbed for AI research, especially with the introduction of toolkits [CKY⁺18]. We believe that DQNs can also be used to play D&D; while superficially the same, major differences in what the agent can know make this a distinct challenge.

The field of *interactive narrative* concerns itself with the creation of digital interactive experiences in which users create or influence a dramatic storyline through actions, either by assuming the role of a character in a fictional virtual world, issuing commands to computer-controlled characters, or directly manipulating the fictional world state [RB13]. Interactive narratives sometimes make use of an *Experience Manager*—also called a *Drama*

Manager—an intelligent, omniscient, and disembodied agent that monitors the virtual world and intervenes to drive the narrative forward according to some model for quality of experience. An experience manager progresses the narrative by intervening in the fictional world, typically by directing computer-controlled characters in how to respond to the users actions. Riedl and Bultiko [RB13] give a high-level overview of some of the techniques that have been attempted. Reinforcement-learning-based approaches to drama management include [BRN⁺07] and [HR16].

Interactive narratives share a lot of similarities with TRPGs. However, players do not describe their actions in natural language but use point-and-click action interfaces to interact with the world. In some instances, the player can engage in dialogue with NPCs through unconstrained natural language [MS03]. Nonetheless, NPCs in interactive narratives are constrained to a fixed and pre-specified repertoire of actions and dialogue. In this paper we focus on the opposite problem of AI agents that *play* TRPGs, and to make the problem more tractable, there is no external evaluator of actions nor Experience Manager.

Contrasting IF playing and drama management, Interactive Fiction *generation* systems use pre-existing resources to develop dynamic IF that adapts to the player’s choices. Systems like Scheherazade-IF [GHLR15] and DINE [CGO⁺17] were strongly influenced by *automated story generation*, giving the user control again, whereas a traditional IF simply has the user discover the preexisting story. Playing a TRPG shares many of the same challenges as being able to automatically generate a story; both story generation and TRPG playing require an agent to select what a character will do next. Automated story generation has a long history of using planning systems [Mee77, Leb87, CCM02, PC09, RY10, WY11] that work in well-defined domains. Recently, machine learning has been used to build story generation systems that automatically acquire knowledge about domains and how to tell stories from natural language corpora [LLUJR13, SG12, RG15, KBT17, GAG⁺17, MAW⁺18]. Our approach draws heavily from neural-network-based approaches.

3 Proposed Approach

Similar to text adventure games, a TRPG’s game state is hidden. However, what makes TRPGs different from text adventure games is the lack of a shared game engine to maintain a ground-truth state of the fictional world and to provide a fixed set of allowable actions. That is, the “game engine” is largely in the heads of the players and each player may have a different understanding of the world state. This makes playing TRPGs more akin to improvisational theater acting [MMR⁺09]. While the Game Master may be considered the maintainer of ground-truth state and an arbiter of what can and cannot be done in the fictional world, the GM’s belief about the state of the world is just one of many and refereeing is mostly restricted to combat and other formulaic parts of the game. Still, one may assume that, just as with the real world, the fictional world does have some rules and conventions, some of which may be explicit, others can be implied. Marie-Laure Ryan named this implication the *principle of minimal departure*, which says that, unless stated otherwise, we are to assume that a fictional world matches our actual world as closely as possible [Rya80]. This means that the fictional world that our agent operates in should have as many similarities to our actual world as we can give it.

This poses a problem though; how can the agent acquire models of the explicit and implicit rules of the fictional world? A standard technique in machine learning is to train a model on a corpus of relevant data. In our case, the most relevant data from which to learn a model is likely to be stories from the particular genre of fictional world our agent will be inhabiting. While it is possible to learn a model of likely event sequences (i.e. machine-learned story generation models [RG15, MAW⁺18, KBT17, FLD18]), recurrent neural networks maintain state as hidden neural network layers, which are limited in the length of their memory and do not explicitly capture the underlying reason why certain events are preceded by others. This is essential because the other, human players may make choices that are very different from sequences in a training corpus—what are referred to as “out of distribution”—and are capable of remembering events and state information for long periods of time. Because of the principle of minimal departure, story generation models also fail to capture details that we take for granted in our own lives—details that are too mundane to mention in stories, such as the affordances of objects. For example, the system would be unable to understand why a cow can be hurt but a cot can’t no matter how much weight you put on it.

Our proposal has two parts. First, we propose a method for acquiring models of the fictional world by blending commonsense, overarching rules about the real world with automated methods that can extract relevant genre information from stories. Second, we propose a reinforcement learning technique based on Deep *Q* Networks that can learn to use these models to interact with human TRPG players. Our proposed agent works as follows. It first converts any human player’s declaration of action—a natural language sentence—into an *event*, which

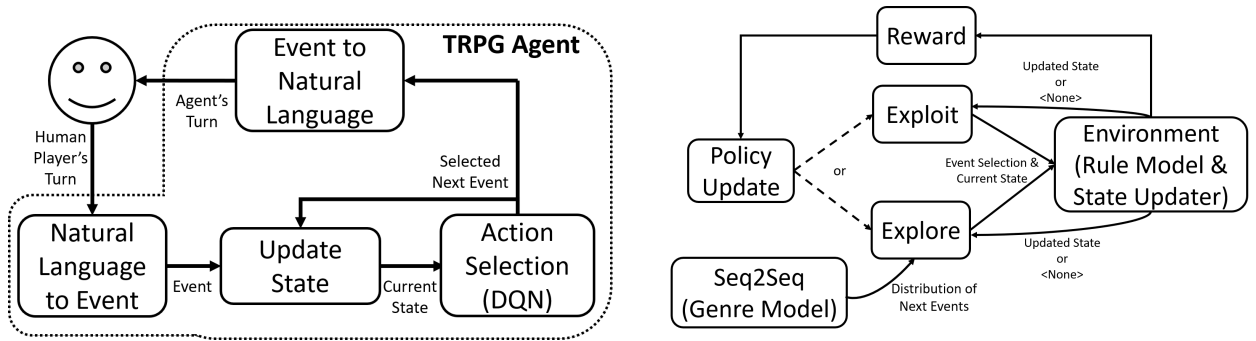


Figure 1: (a) The entire pipeline of agent once the reinforcement learner is trained, and (b) details of the reinforcement learner while training.

is an abstract sentence representation that is easier for AI systems to work with. We will describe the event representation in Section 3.1. This event is used to update the agent’s belief about the state of the fictional world. Once the state is updated, the agent takes its turn, selecting a new event using the deep reinforcement learner. The state is updated again and the agent’s event is convert back into natural language so that the human player can read what the agent did. This pipeline can be seen in Figure 1(a).

The training method is shown in Figure 1(b). While the DQN is exploring during training, the previous event in the story is passed into a Sequence-to-Sequence LSTM [SVL14] that is trained on data from the genre we selected. The Seq2Seq network generates a distribution over possible subsequent events according to our model of genre expectations. A set of rules filters the list of events, keeping only events that could occur given the current state of the game. The agent chooses to exploit its policy model or explore randomly, and once a valid event is picked, the state is updated. Because we have a rule model, we can conduct multi-step lookahead, wherein the agent explores several steps into the future before using the reward to update the policy. Each event that is picked should bring the agent closer to its goal for the campaign. The goal in this case is a genre-appropriate pre-defined event that we select.

3.1 The Agent’s Model of the Fictional World

In this section we describe the two-part model of the fictional world that the agent has access to in order to select actions and update its understanding of the current state of the game world.

3.1.1 The Genre Expectation Model

Given a corpus of stories from a genre related to the fictional world that the agent will inhabit, we train a *genre expectation model*. This model provides the probability, according to the genre-specific corpus, that certain events will happen after other events. Specifically, we model genre expectations as a sequence-to-sequence LSTM [SVL14], a type of recurrent neural network. However, instead of training on natural language sentences, we first convert sentences to events.

An event is a tuple $\langle s, v, o, n, p \rangle$ where v is the verb of the sentence, s is the subject of the verb, o is the direct object of the verb, n is the noun of a propositional phrase (or indirect object, causal complement, or any other significant word), and p is the preposition. Martin et al. [MAW⁺18] found that representations similar to this assist with the accuracy of models trained on stories. In this work, we add the p element, which Pichotta and Mooney [PM16] also found to be helpful for event representations. These words are extracted from the original sentence, stemmed, and then *generalized* or looked up in their respective databases. Verbs are abstracted using VerbNet [KS05], which provides a category—or *class*—for each verb. All other words are queried in a hierarchical lexicon called WordNet [Mil95], and the generalized word/*Synset* is taken from two levels up in the hierarchy (i.e. the grandparent). Adding the preposition to the event enables us to make closer comparisons between possible syntactic constructions within a VerbNet class since we would not have access to the original sentence’s syntax.

The story corpus for the genre is pre-processed such that each clause is transformed into an event. Using these events, we train the Seq2Seq model to predict the probability of an event given a previous event. This becomes the agent’s genre expectation model, giving the agent a pool of likely actions to choose from during its turn. However, it does not not guarantee that they are valid or logical actions though, nor would it be able to keep track of long-term dependencies.

3.1.2 Commonsense Rules Model

To help the agent with selecting appropriate events/actions, we acquire a second model of general, commonsense rules about the real world. The purpose of this model is to (a) prune out candidate events that would not work for the current state of the game, and (b) allow the agent to do lookahead planning to determine how current actions might affect future world states.

The rules are acquired from a set of semantic facts we get from VerbNet. In VerbNet, each verb class has a set of *frames*. Each frame is determined by a grammatical construction that this verb can be found in. Within a frame, the syntax is listed, along with a set of semantics. The semantics specify what roles/entities are doing what in the form of *predicates*. For example, VerbNet would tell us that the sentence “Lily screwed the handle to the drawer” yields the following predicates:

- CAUSE(Agent,Event)
- TOGETHER(end(Event), Patient, Co-Patient)
- ATTACHED(end(Event),Patient, Instrument)
- ATTACHED(end(Event),Co-Patient, Instrument)

where *Lily* is the Agent, *the handle* is the Patient, *the drawer* is the Co-Patient, and *screw* is the Instrument. In other words: Lily caused the event, and at the end of the event, the screw attached the drawer and the handle together.

Based on the principle of minimal departure, our agent assumes that when an event occurs, the frame’s predicates hold, acting as the agent’s knowledge about the actual world. This is reasonable because the frame semantics are relatively high-level and can occur in a variety of scenarios. Whereas the state of the genre expectation model is latent, we can use the facts generated by applying commonsense rules to maintain explicit beliefs about the world that persist until new facts replace them. That is, the drawer and handle will remain attached until such time that another verb class indicates that they are no longer attached. This is important because the agent’s belief statement won’t be tied to a limited, probabilistic window of history maintained by the genre expectation model.

However, the predicates currently provided by VerbNet frames are insufficient for our purposes. We augment VerbNet by breaking down predicates that required more detail. All of the predicates are either considered “core predicates”—where they cannot be broken down further—or are given other existing predicates to form *preconditions* and *post-conditions*. Preconditions are conditions that must be true in the world prior to the verb frame being enacted. Post-conditions—or effects—are facts about the world that hold after a verb frame has finished being enacted. This information would not be learned by a recurrent neural network.

We use the preconditions to filter out any actions proposed by the genre expectation model that are not consistent with the current state of the world. Once an action is selected, we use the post-conditions to update the agent’s belief state about the world. Magerko et al. made use of pre- and post-conditions for actions so that the individual agents separate from the Game Manager in their game kept the story consistent [MLA⁺04]. Similarly, Clark et al. [CDT18] broke VerbNet semantics into pre- and post-conditions.

3.2 Reinforcement Learning for TRPGs

Reinforcement learning (RL) is a technique whereby an agent learns a *policy*, mapping states to actions that maximize expected future reward. A reward function gives a value indicating how good or bad different states are, and it can be sparse—meaning that a reward is not provided very often. RL agents learn the policy incrementally by trial and error, attempting to find correlations between states s_i and future expected reward, $Q(s_i)$. Deep Q networks learn to approximate the expected future reward for states with a neural network. Deep Q network agents have been shown to be able to play complicated games such as Atari video games [MKS⁺13] and text adventures with limited action spaces [NKB15, HZMM18, YCS⁺18].

Our event representation turns an infinite number of actions (any natural language sentence) into a large, but finite action space. Still, we cannot perform exhaustive trial and error learning while training a deep Q network to play D&D. The genre expectation model provides a beam of highly probable events that are consistent with the genre. The commonsense rule model filters events and also allows acts as a transition function, helping the agent to search through possible future states for those that return the highest reward, which will, in turn, help the agent converge faster in situations where the reward (e.g. reaching a particular state) is sparse. Details on the flow of information through the DQN can be seen in Figure 1(b).

4 Future Work

One of the outstanding limitations of our current proposal is the reliance on a reward function. For the near future, rewards are based on quest completion, although that is only one aspect of the tabletop roleplaying game experience. Quest completion is a sparse reward, which is one of the reasons why the commonsense rules will be useful in allowing the agent to lookahead since most states will not provide any reward signal. In the future, we will need to identify or learn more complete reward functions.

Future versions of the system can be created to learn what rules are broken by the user and remain consistent with them. This will require the ability for the agent to identify broken rules and then remove them from its processing of potential actions to take. It might also import other genre models. For example, if the user has raised Vinay from the dead, the agent now knows that after a character dies, they are no longer just removed from the story but can be reanimated. It might also integrate a horror genre that includes zombies. For now, we will start with a strict set of rules that the agent must obey when it is playing the game, and the agent will work within one genre at a time.

5 Conclusions

As game-playing AI research progresses, we argue that TRPGs like D&D are an appropriate next challenge. TRPGs are unique from other games in that they have an infinite selection of actions, have a partially-visible world, contain hidden states, use intrinsic reward, do not have explicit progress markers, and are cooperative. We outlined a subproblem of TRPGs which focuses less on character stats, which we already know computers to handle well, but also simplifies the problem slightly by eliminating the refereeing of rules. TRPGs are unlike text adventure games in that the players have more agency in affecting the story, but it is also unlike Drama Management where the system gives the player some control over the story but still has the final say. TRPG players are more similar to collaborative automated story generators in this way.

To create an AI that plays this modified TRPG, we proposed that the agent has a model of the world that is a combination of rules about our actual world and a concept of what events usually occur within similar fictional worlds. The agent is then trained to use the model through deep Q -learning, which has been successful in playing games. By sharing our plans for our TRPG player, we hope that we inspire other AI researchers to look into this unique space of games.

Acknowledgments

This work is supported by DARPA W911NF-15-C-0246. The views, opinions, and/or conclusions contained in this paper are those of the authors and should not be interpreted as representing the official views or policies, either expressed or implied of the DARPA or the DoD.

References

- [BRN⁺07] Sooraj Bhat, David L. Roberts, Mark Nelson, Charles Isbell, and Michael Mateas. A globally optimal algorithm for TTD-MDPs. In *Proceedings of the 6th International Joint Conference on Autonomous Agents and Multiagent Systems*, 2007.
- [CCM02] M. Cavazza, F. Charles, and S. Mead. Planning characters' behaviour in interactive storytelling. *Journal of Visualization and Computer Animation*, 13:121–131, 2002.
- [CDT18] Peter Clark, Bhavana Dalvi, and Niket Tandon. What Happened? Leveraging VerbNet to Predict the Effects of Actions in Procedural Text. *arXiv:1804.05435*, 2018.
- [CGO⁺17] Margaret Cychosz, Andrew S. Gordon, Obiageli Odimegwu, Olivia Connolly, Jenna Bellasai, and Melissa Roemmele. Effective scenario designs for free-text interactive fiction. In Nuno Nunes, Ian Oakley, and Valentina Nisi, editors, *Interactive Storytelling*, pages 12–23. Springer International Publishing, 2017.
- [CKY⁺18] Marc-Alexandre Cote, Akos Kadar, Xingdi (Eric) Yuan, Ben Kybartas, Tavian Barnes, Emery Fine, James Moore, Matthew Hausknecht, Layla El Asri, Mahmoud Adada, Wendy Tay, and Adam Trischler. TextWorld: A Learning Environment for Text-based Games. In *Computer Games Workshop at ICML/IJCAI 2018*, pages 1–29, June 2018.

- [FLD18] A. Fan, M. Lewis, and Y. Dauphin. Hierarchical Neural Story Generation. *arXiv:1805.04833*, 2018.
- [GA74] Gary Gygax and Dave Arneson. *Dungeons & Dragons*, 1974.
- [GAG⁺17] Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N. Dauphin. Convolutional Sequence to Sequence Learning. *arXiv:1705.03122*, 2017.
- [GHLR15] Matthew Guzdial, Brent Harrison, Boyang Li, and Mark Riedl. Crowdsourcing open interactive narrative. In *10th International Conference on the Foundations of Digital Games (FDG 2015)*, 2015.
- [HA04] Brian Hlubocky and Eyal Amir. Knowledge-gathering agents in adventure games. In *AAAI-04 workshop on Challenges in Game AI*, 2004.
- [HR16] Brent Harrison and Mark O Riedl. Learning from stories: Using crowdsourced narratives to train virtual agents. In *Proceedings of the 2016 AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, 2016.
- [HZMM18] Matan Haroush, Tom Zahavy, Daniel J. Mankowitz, and Shie Mannor. Learning How Not to Act in Text-Based Games. In *Workshop Track at ICLR 2018*, pages 1–4, 2018.
- [KBT17] Ahmed Khalifa, Gabriella AB Barros, and Julian Togelius. Deeptingle. *arXiv:1705.03557*, 2017.
- [KKKR17] Bartosz Kostka, Jaroslaw Kwiecieli, Jakub Kowalski, and Pawel Rychlikowski. Text-based adventures of the Golovin AI agent. *2017 IEEE Conference on Computational Intelligence and Games, CIG 2017*, pages 181–188, 2017.
- [KS05] Karen Kipper-Schuler. *VerbNet: A Broad-Coverage, Comprehensive Verb Lexicon*. PhD thesis, University of Pennsylvania, 2005.
- [LC16] Guillaume Lample and Devendra Singh Chaplot. Playing FPS games with deep reinforcement learning. *CoRR*, abs/1609.05521, 2016.
- [Leb87] Michael Lebowitz. Planning stories. In *Proceedings of the 9th Annual Conference of the Cognitive Science Society*, pages 234–242, 1987.
- [LLUJR13] Boyang Li, Stephen Lee-Urban, George Johnston, and Mark O. Riedl. Story generation with crowd-sourced plot graphs. In *Proceedings of the 27th AAAI Conference on Artificial Intelligence*, Bellevue, Washington, July 2013.
- [LvL01] John Laird and Michael van Lent. Human-level AI’s killer application: Interactive computer games. *AI Magazine*, 22(2):15–25, 2001.
- [MAW⁺18] Lara J. Martin, Prithviraj Ammanabrolu, Xinyu Wang, William Hancock, Shruti Singh, Brent Harrison, and Mark O. Riedl. Event Representations for Automated Story Generation with Deep Neural Nets. In *Thirty-Second AAAI Conference on Artificial Intelligence (AAAI-18)*, pages 868–875, New Orleans, Louisiana, 2018.
- [Mee77] James R. Meehan. TALE-SPIN: An interactive program that writes stories. In *Proceedings of the 5th International Joint Conference on Artificial Intelligence*, pages 91–98, 1977.
- [Mil95] George A. Miller. WordNet: a Lexical Database for English. *Communications of the ACM*, 38(11):39–41, 1995.
- [MKS⁺13] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. Playing Atari with Deep Reinforcement Learning. *arXiv:1312.5602*, 2013.
- [MLA⁺04] Brian Magerko, John E. Laird, Mazin Assanie, Alex Kerfoot, and Devvan Stokes. AI characters and directors for interactive computer games. *Proceedings of the Nineteenth National Conference on Artificial Intelligence, Sixteenth Conference on Innovative Applications of Artificial Intelligence*, pages 877–883, 2004.

- [MMR⁺09] Brian Magerko, Waleed Manzoul, Mark Riedl, Allan Baumer, Daniel Fuller, Kurt Luther, and Celia Pearce. An empirical study of cognition and theatrical improvisation. In *Proceedings of the Seventh ACM Conference on Creativity and Cognition*, pages 117–126, New York, NY, USA, 2009. ACM.
- [MS03] Michael Mateas and Andrew Stern. Integrating plot, character, and natural language processing in the interactive drama Façade. In *Proceedings of the 1st International Conference on Technologies for Interactive Digital Storytelling and Entertainment*, 2003.
- [NKB15] Karthik Narasimhan, Tejas Kulkarni, and Regina Barzilay. Language Understanding for Text-based Games Using Deep Reinforcement Learning. In *EMNLP*, page 10, 2015.
- [Ope] OpenAI. OpenAI DOTA 2 1v1 bot, 2017.
- [PC09] Julie Porteous and Marc Cavazza. Controlling narrative generation with planning trajectories: the role of constraints. In *Proceedings of the 2nd International Conference on Interactive Digital Storytelling*, pages 234–245, 2009.
- [PM16] Karl Pichotta and Raymond J Mooney. Learning Statistical Scripts with LSTM Recurrent Neural Networks. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, pages 2800–2806, 2016.
- [RB13] Mark O. Riedl and Vadim Bulitko. Interactive narrative: An intelligent systems approach. *AI Magazine*, 34(1):67–77, Spring 2013.
- [RG15] Melissa Roemmele and Andrew S. Gordon. Creative help: A story writing assistant. In *Proceedings of the Eighth International Conference on Interactive Digital Storytelling*, 2015.
- [RY10] Mark O. Riedl and R. Michael Young. Narrative planning: Balancing plot and character. *Journal of Artificial Intelligence Research*, 39:217–268, 2010.
- [Rya80] Marie-Laure Ryan. Fiction, non-factuals, and the principle of minimal departure. *Poetics*, 9(4):403–422, 1980.
- [SG12] Reid Swanson and Andrew S. Gordon. Say Anything: Using Textual Case-Based Reasoning to Enable Open-Domain Interactive Storytelling. *ACM Transactions on Interactive Intelligent Systems*, 2(3):1–35, 2012.
- [SHM⁺16] David Silver, Aja Huang, Chris J. Maddison, Arthur Guez, Laurent Sifre, George van den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, Sander Dieleman, Dominik Grewe, John Nham, Nal Kalchbrenner, Ilya Sutskever, Timothy Lillicrap, Madeleine Leach, Koray Kavukcuoglu, Thore Graepel, and Demis Hassabis. Mastering the game of Go with deep neural networks and tree search. *Nature*, 529(7587):484–489, 2016.
- [SVL14] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112, 2014.
- [WY11] Stephen Ware and R. Michael Young. CPOCL: A narrative planner supporting conflict. In *Proceedings of the 7th AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, 2011.
- [YCS⁺18] Xingdi Yuan, Marc-Alexandre Côté, Alessandro Sordani, Romain Laroche, Remi Tachet Des Combes, Matthew Hausknecht, and Adam Trischler. Counting to Explore and Generalize in Text-based Games. *arXiv:1806.11525*, 2018.