

Improvisational Computational Storytelling in Open Worlds

Lara J. Martin, Brent Harrison, and Mark O. Riedl

School of Interactive Computing, Georgia Institute of Technology, Atlanta, GA
{lara.martin, brent.harrison, riedl}@cc.gatech.edu

Abstract. Improvisational storytelling involves one or more people interacting in real-time to create a story without advanced notice of topic or theme. Human improvisation occurs in an open-world that can be in any state and characters can perform any behaviors expressible through natural language. We propose the grand challenge of computational improvisational storytelling in open-world domains. The goal is to develop an intelligent agent that can sensibly co-create a story with one or more humans through natural language. We lay out some of the research challenges and propose two agent architectures that can provide the basis for exploring the research issues surrounding open-world human-agent interactions.

Keywords: brave new ideas • intelligent narrative technologies • computational improvisation • interactive narrative

1 Introducing Computational Improvisation

Storytelling has been of interest to artificial intelligence researchers since the earliest days of the field. Artificial intelligence research has addressed story understanding, automated story generation, and the creation of real-time interactive narrative experiences. Specifically, *interactive narrative* is a form of digital interactive experience in which users create or influence a dramatic storyline through actions 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 [1]. Interactive narrative requires an artificial agent to respond in real time to the actions of a human user in a way that preserves the context of the story and also affords the user to exert his or her intentions and desires on the fictional world. Prior work on interactive narrative has focused on *closed-world domains*—a virtual world, game, or simulation environment constrained by the set of characters, objects, places, and the actions that can be legally performed. Such a world can be modeled by finite AI representations, often based on logical formalizations. In this paper, we propose a grand challenge of creating artificial agents capable of engaging with humans in improvisational, real-time storytelling in *open worlds*.

Improvisational storytelling involves one or more people constructing a story in real time without advanced notice of topic or theme. Improvisational storytelling is often found in improv theatre, where two or more performers receive suggestions of theme from the audience. Improvisational storytelling can also happen in informal

settings such as between a parent and a child or in table-top role-playing games. While improvisational storytelling is related to interactive narrative, it differs in three significant ways. First, improvisational storytelling occurs in open worlds. That is, the set of possible actions that a character can perform is the space of all possible thoughts that a human can conceptualize and express through natural language. Second, improvisational storytelling relaxes the requirement that actions are strictly logical. Since there is no underlying environment other than human imagination, characters' actions can violate the laws of causality and physics, or simply skip over boring parts. However, no action proposed by human or agent should be a complete non sequitur. Third, character actions are conveyed through language and gesture.

In this paper we explore the challenges from and potential solutions to creating computational systems that can engage with humans in improvisational storytelling. We envision a system in which humans control some characters in an open world, while some are controlled by artificial intelligence. Beyond entertainment, computational improvisational storytelling unlocks the potential for a number of serious applications. Computational improvisational storytelling systems could engage with forensic investigators, intelligence analysts, or military strategists to hypothesize about crimes or engage in creative war-gaming activities. Virtual agents and conversational chatbots can also create a greater sense of rapport with human users by engaging in playful activities or gossip. Successful development of an artificial agent capable of engaging with humans in open-world improvisational storytelling will demonstrate a human-level ability to understand context in communication. It will also provide an existence proof that artificial intelligence can achieve human-like creativity.

2 Background

2.1 Interactive Narrative

Riedl and Bulitko [1] give an overview of AI approaches to interactive narrative. The most common form of interactive narrative involves the user taking on the role of the protagonist in an unfolding storyline. The user can also be a disembodied observer—as if watching a movie—but capable of making changes to the world or talking to the characters. A common solution, first proposed by Bates [2] is to implement a *drama manager*. A drama manager is an intelligent, omniscient, and disembodied agent that monitors the virtual world and intervenes to drive the narrative forward according to some model of quality of experience. An *experience manager* [3] is a generalization of this concept, recognizing the fact that not all narratives need to be dramatic, such as in the case of education or training applications.

There are many AI approaches to experience management. One way is to treat the problem of story generation as a form of search such as planning [3–5], adversarial search [6, 7], reinforcement learning [8], or case-based reasoning [9, 10], although planning is still appropriate for games since they are closed systems [11]. All of the above systems assume an *a priori*-known domain model that defines what actions are available to a character at any given time.

Closed-world systems can sometimes appear open. *Façade* [12] allows users to interact with virtual characters by freely inputting text. This gives the *appearance* of open communication between the human player and the virtual world; however, the system limits interactions by assigning the natural language input to dramatic beats that are part of the domain model. Open-world story generation attempts to break the assumption of an *a priori*-known domain model. *Scheherazade-IF* [13] attempts to learn a domain model in order to create new stories and interactive experiences in previously unknown domains. However, once the domain is learned, it limits what actions the human can perform to those within the domain model. *Say Anything* [14] is a textual case-based-reasoning story generator, meaning that it operates in the space of possible natural language sentences. It responds to the human user by finding sentences in blogs that share a similarity to human-provided sentences, but consequently tends to fail to maintain story coherence without human intervention.

2.2 Improv Theatre

Humans have the ability to connect seemingly unrelated ideas together. If a computer is working together with a user to create a new story, the AI must be prepared to handle anything the human can think of. Even when given a scenario that appears constrained, people can—and will—produce the unexpected. Magerko et al. [15] conducted a systematic study of human improv theatre performers to ascertain how they are able to create scenes in real time without advanced notice of topic or theme. The primary conclusion of this research is that improv actors work off of a huge set of basic scripts that compile the expectations of what people do in a variety of scenarios. These scripts are derived from common everyday experiences (e.g., going to a restaurant) or familiar popular media tropes (e.g., Old West shoot out). Magerko and colleagues further investigated possible computational representations of scripts used in improv [16], and how improv actors create and resolve violations in scripts [17].

3 Open-World Improvisational Storytelling

In order to push the boundaries of AI and computational creativity we argue that it is essential to explore open-world environments because (a) we know that humans are capable of doing so, especially with training (actors, comedians, etc.), and (b) natural language interaction is an intuitive mode of human-computer interaction for humans that is not constrained to finite sets of well-defined actions. Once the mode of interaction between a human and an artificial agent is opened up to natural language, it would be unnatural and ultimately frustrating for the human to restrict their vocabulary to what the agent can understand and respond sensibly to. An intelligent agent trained to work from within a closed world will struggle to come up with appropriate responses to un-modeled actions. On the other hand, limiting the user's actions and vocabulary also limits the user's creativity.

There are two general problems that must be addressed to achieve open-world improvisational storytelling. First, an intelligent improvisational agent must have a set of scripts comparable in scope to that held by a human user. This is in part addressed

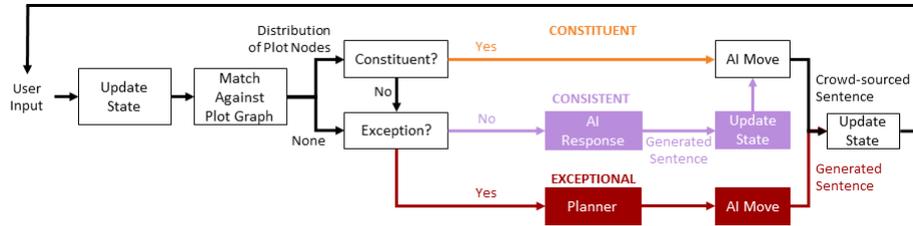


Fig. 1. The proposed plot graph system’s architecture.

by systems, such as *Scheherazade* [18] which learns scripts from crowdsourced example stories, or various projects learning scripts from internet corpora [19, 20]. We loosely define a *script* as some expectation over actions. To date, no system has demonstrated that it has learned a comprehensive set of scripts; however, once a comprehensive set of scripts exists, these scripts can be used to anticipate human actions that are consistent with the scripts and generate appropriate responses.

Second, an intelligent improvisational agent must be able to recognize and respond to *off-script* actions. This means that the agent will need to generate new, possibly off-script actions of its own in order to respond to the player in a seemingly non-random way. The reasons why a human goes off script can be technical—the human’s script does not exactly match the agent’s script for the same scenario—or because the human wishes to express creative impulses or test the boundaries of the system.

Since humans normally tend to work off of some sort of script while improvising, whether it is explicit or not, the AI also needs to relate user utterances to a script through natural language understanding. Keeping track of a script is a matter of comparing the meanings of human utterances—or *semantics*—which is an open research question. Given language’s nearly infinite possibilities, it is very unlikely that two people would use the exact same words or syntax to express the same idea. It is just as unlikely that a user would create a similar sentence as the creators of the agent would. Beyond understanding the meaning of individual sentences, there is still the matter of what the semantics of the sentence mean within the context of the entire story—also known as the *pragmatics*—since context is important to maintaining coherence in a conversation.

In the remaining sections, we will introduce two potential approaches. The first uses script representations closely aligned with observations of improv actors. The second uses neural networks trained on a corpus of stories.

4 Plot Graph Approach

We present a first attempt at creating a computational architecture for maintaining a coherent story context when co-creating with a human user. We acknowledge that open-world improvisation will not be solved until we address many open research challenges. Our goal is to offer an initial conjecture about how an improvisational agent might be built, which can be expanded upon as research challenges become solved.

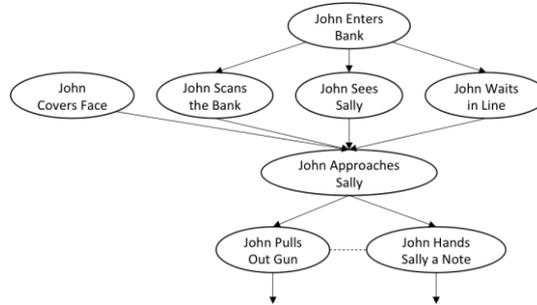


Fig. 2. A section of the robbery plot graph used in our system, where solid arrows represent temporal precedence and dotted lines represent mutual exclusion between plot event nodes.

The proposed system architecture is shown in Fig. 1. First, the user enters in a line of text to narrate the action that they want the main character to take. This text is compared against the script, in this case represented as a *plot graph* (Section 4.1). The natural language processing happens in two stages: interpreting the semantic content in order to track the world state (Section 4.2) and determining the constituency of the user’s action (Section 4.3). The agent employs different strategies for responding to sentences based on whether they describe actions that are constituent, consistent, or exceptional. Once the AI produces a sentence, the entire process repeats.

4.1 Introduction to Plot Graphs

For our work, we assume a script representation; in particular, one called a *plot graph*. Plot graphs have been found to be effective for interactive narrative and story generation [6–8, 13, 18]. We use the representation developed by Li et al. [18], which facilitates script learning from crowdsourced narrative examples. A plot graph is a script representation that compactly describes a space of possible stories (and thus expectations about stories) that can be told about a specific scenario such as going to a restaurant or robbing a bank. A plot graph is a directed acyclic graph where the nodes in the plot graph are events. One type of edge represents temporal precedence relationships between events. For example, consider the plot graph in Fig. 2, which shows a fragment of a plot graph for robbing a bank. The plot graph node *<John Enters the Bank>* is connected to *<John Scans the Bank>*, which means that the former event must be completed before the latter would be expected to begin. These relationships, however, are not strict causal relationships. The node *<John Covers Face>*, for example, is also connected to *<John Approaches Sally>* but has no parent node. This means that it can be executed at any time as long as it occurs before John approaches Sally. A second type of link between nodes represents mutual exclusivity of events, where the occurrence of one event predicts the absence of the other. Mutual exclusions encode branches in the plot where a choice leads to different variations of the scenario. Each plot graph node contains a set of semantically-similar sentences, describing the same event in different ways. A plot graph can be used to generate different legal sequences of events. The plot graph in Fig 2. was learned from data and algorithms from Li et al. [18].

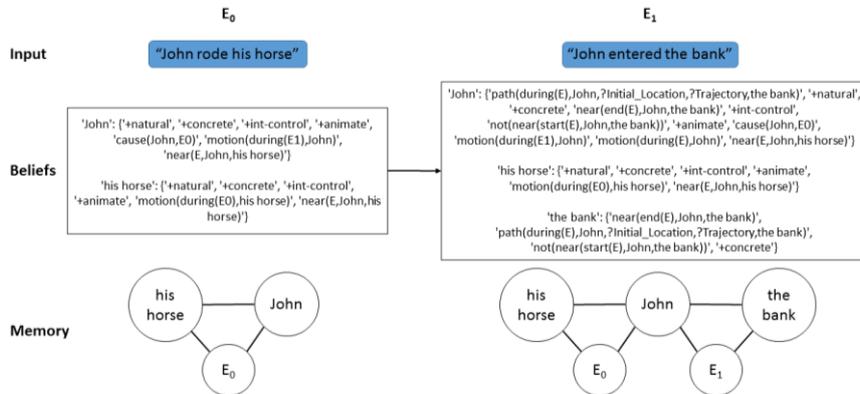


Fig. 3. An example of the two parts of the state where the first event is “John rode his horse” (E_0), followed by the second event: “John entered the bank” (E_1). On the top half of the figure are the beliefs of the AI shown after E_0 and E_1 , respectively. The bottom half of the figure shows the AI’s memory during these events.

4.2 Maintaining World State

One of the challenges of open-world improvisational storytelling is representing and maintaining a world state without making too many constraining assumptions about the entities and relations between entities that can occur in the world. Furthermore, new entities, objects, and places can be created at any time. We propose that the AI’s internal world state contains two aspects: the AI’s current knowledge about the world’s usable items (beliefs), and the AI’s memory of the story.

The AI’s belief system about world state must ground each user or AI turn in terms of the semantics of the action expressed in natural language. One solution is to ground all sentences using *VerbNet* [21], an ontology of verbs and their syntax-dependent semantics created by linguists to serve as a domain-independent verb lexicon. *VerbNet* consists of *frames* for sets of verbs that are semantically and generally syntactically equivalent. Frames contain rules for how to label entities playing roles in the sentences (e.g., “John rode his horse to the bank” infers that “John” and “his horse” are now at “the bank”), predicate-like facts about those roles (e.g., “John rode his horse” means that “John” and “his horse” moved), and limiting factors for which entities can fill roles (e.g., the verb “ride” requires an animate object).

We are augmenting *VerbNet* to include predicates about *accessibility* and *proximity*. Entities that are accessible exist in the world and entities that are proximate to the AI agent can be directly manipulated. When the user or the agent takes a turn, grounded predicates are added to the agent’s belief state, which continues to grow at every turn unless something becomes inaccessible (i.e., it dies, it disappears, it is eaten, etc.). See the top portion of Fig. 3 for an example of beliefs.

The AI’s memory allows the agent to organize concepts based on recency and spatiality. We propose to use the *event-indexing model* [22], a psychology framework that is meant to explain how people comprehend new information and store it in their long-term memory. Event indexing has been modified previously for artificial intelligence and story comprehension to model a readers’ memory [23, 24]. The AI

maintains a memory graph of what nouns (entities) were mentioned during which turns and what is spatially proximate. Entities are connected to a special node representing the event, E_n , as well as to any other entity referenced by the event's sentence. The salience of objects can be computed as the distance in the memory graph between two objects. See the bottom portion of Fig. 3 for an example of an event-indexing memory structure that corresponds to the world state beliefs.

4.3 Responding to the User

Recall that improvisational storytelling involves identifying the script for a situation and then breaking that script. Therefore, one of the first things a computational improvisational agent must do is to identify whether the user is attempting to follow a script or break it. Riedl et al. [4] established a classification for how user actions relate to the sense of continuity and coherence in interactive narratives:

- *Constituent*: User actions that meet the expectation of the AI. In the case of improvisational storytelling, a constituent action is one predicted by the AI's script.
- *Consistent*: User actions that do not meet the expectation of the AI but do not prevent the AI from continuing to execute according to its script.
- *Exceptional*: User actions that do not meet the expectation of the AI and exclude the possibility of the AI continuing to execute according to its script.

An improvisational agent must determine if a user action is constituent, consistent, or exceptional. Consistent actions should be responded to before continuing with actions recommended by a script. Exceptions have been the subject of prior research [3, 4, 25]. Once an exceptional user action is identified, we need to have a system that can respond in a sensible matter via planning. We need a way to turn the action that was decided as an appropriate next step into a set of semantic units, and then translate the semantics into some sort of grammatical sentence. Natural language generation is not yet a solved problem, let alone generating creative sentences.

Constituent Branch. A constituent action indicates that the user is likely to be following the script. The constituent response strategy is shown in orange in Fig. 1. Our check for constituency is straightforward: The agent checks to see if the user's sentence matches against one of the plot events that can directly succeed the most recently-executed plot event. If the user's action is constituent, the agent can follow up by selecting a successor plot event from the plot graph (if the plot event belongs to an AI-controlled character). To respond to the user, the agent can choose any existing sentence in the cluster of sentences associated with the selected plot graph node.

Consistent Branch. Consistent actions are those that do not move the script forward but also do not prevent it from moving forward. The agent's strategy for consistent actions is shown in purple in Fig. 1. If the agent fails to match the user's sentence to the plot graph and the action is not deemed an exception, the agent continues the story by generating an off-script response. While there are many techniques that can be used to select off-script responses, our proposal is to generate a response by identifying objects with high-salience from its "memory", selecting one that is

accessible, and then determining which actions can be performed with said object. One way to determine which verbs can be performed on an entity is to query ConceptNet 5 [26], a commonsense knowledgebase that has a large number of facts about objects commonly found in the real world and what they are used for. The actions that can be performed on the object are looked up in VerbNet, and all possible sentences are generated from the verb frame's syntax templates by filling roles. Any sentence generated in this way can be selected randomly or ranked according to likelihood. The likelihood of a sentence can be computed by constructing a language model over a large corpus such as Wikipedia or Google's Project Gutenberg that estimates the how likely combinations of verbs and nouns are to co-occur in the English language.

Exceptional Branch. Exceptions occur when the user's action causes the world to enter a state in which no successor in the plot graph can be executed because one or more preconditions of each successor plot point is contradicted. Exceptions can also occur when future plot events that must occur likewise have their preconditions contradicted. For example, a character expected to contribute to the script is absent, a necessary object is missing, or the user is in the wrong place. The agent's strategy for handling exceptions is shown in red in Fig. 1. The improvisational agent must still act, and one strategy is to "repair" the script by finding another action, or sequence of actions, that is not part of the plot graph but restores the world state such that a subsequent plot point can execute. This repair process can be modeled as a planning problem: the task of searching for a sequence of actions that transforms the world state into one in which a goal situation holds. Planning has been applied to repairing stories represented as partial-order plans [3, 4] and stories represented as petri nets [25].

Planning can be used to repair plot graphs as well. The goal situation is any state in which the preconditions of a successor plot point or descendant of a successor holds. By finding a sequence of actions to be performed by the user and AI-controlled entities, the plot graph is restored and able to continue as normal; the planned sequence becomes a branch of the plot graph. However, there is at least one remaining open challenge. The space of possible actions, being all actions that can be expressed in language, is very large, and the complexity of search is proportional to the branching factor. Therefore, despite work on planning with language in closed worlds [27], the complexity of search through language in an open-world would have a very large, if not infinite, branching factor. Even abstracting actions into *VerbNet* frames results in a branching factor in the thousands (number of frames in the ontology times the number of ways roles can be instantiated with known characters and objects).¹ Fortunately, most repairs are likely to require a sequence of one or two actions. Sampling-based planning algorithms such as Monte Carlo Tree Search may be adapted to story repair.

It is possible that no repair is possible, meaning the planner fails to find a sequence of actions that transforms the world into a state where the plot graph can continue executing. This may be due to non-reversible actions (e.g., an object is destroyed)

¹ This analysis excludes the possibility of creating new objects.

performed by the user or due to the search failing because of the size of the search space and the need to respond within a small, fixed amount of time. In this case, new strategies will be required, such as switching to an emergent, reactionary storytelling mode such as that used to generate the response in the consistent branch.

4.4 Limitations

The proposed architecture addresses the challenges that we put forth earlier in this paper; however, there are several limitations that must be mentioned. One limitation is that this technique assumes the presence of a plot graph to act as a script. While *Scheherazade* has the ability to learn plot graphs from crowdsourced stories, it will be problematic to assume that the system will have access to all possible stories that a user may want to tell. Furthermore, the learned plot graphs may not match the scripts held in human users' heads, so the user may perform actions not in the AI's plot graph or skip over events deemed irrelevant or uninteresting. To handle greater stochasticity of human behavior, it will be advantageous to convert the plot graph into a dynamic probabilistic graph with skip-transitions, allowing the AI to jump to the most appropriate event in the plot graph. Often in improvisational storytelling, one would move seamlessly from one script to another or blend elements of several scripts. A more complete system would require the ability to recognize if the user has changed topics—a common yet not fully solved problem shared with other conversational AI systems—or to merge plot graphs to better explain what the user is trying to do.

Additionally, the performance of the agent is heavily reliant on the performance of the natural language understanding (NLU) techniques used. NLU, especially in the areas of semantic reasoning and pragmatics are still open research problems. Further, *VerbNet* may not be the best technique for tracking semantics, and *ConceptNet* is known to be incomplete. These limitations are enumerated here to recommend research areas likely necessary to move the state of the art in interactive narrative systems toward those fully capable of open-world story improvisation with humans.

5 Neural Network Approach

We previously proposed the use of plot graphs to model expectation in a story and explicitly build up state information using external ontologies and corpora. One alternative way that this could be done is to model expectation using a recurrent neural network (RNN) with long short-term memory (LSTM) nodes to preserve story context. These types of networks can take in a sequence of past events and generate a possibly infinite sequence of new events. During training, RNNs learn a representation of state that is embedded in the network's hidden LSTM layers. As a result, this technique is not as reliant on external ontologies to learn state information. In addition, expectation is innate in an RNN. Each time a new event is presented to the RNN, it will calculate a probability distribution over the expected next events. Thus, script information can be extracted by choosing the most expected event to occur at a given time step [20].

In terms of our initial proposed architecture, this means that an RNN is well suited to handle constituent actions that the user may take, where constituent would mean the user performed an event that the RNN was expecting to see with high probability. However, consistent and exceptional events—events performed by the user that are not high-probability transitions and may also create logical inconsistencies later—may present challenges to RNNs. As with the prior approach, consistent and exceptional events can be handled by turning improvisation into a planning problem.

There has been promising work done using deep reinforcement learning (Deep-RL) to dynamically generate dialogues [28]. Reinforcement learning is a technique for solving planning problems in stochastic and uncertain domains. A reward function provides a measure for how much value the algorithm receives for performing certain actions in certain states; reinforcement learning attempts to maximize expected reward over time. Deep-RL involves the use of a deep neural net to estimate the probability of transitioning from one state to another or the value that will be received in states it has not previously seen. Here, the RNN learns an internal representation of state and uses that in conjunction with an author-supplied reward function to determine the value of generating an event at the current time. Using this framework, these deep neural approaches can handle consistent and exceptional actions. If the user takes *off-script* actions, then the system will still generate events that will maximize its long-term reward. Thus, the system’s behavior is largely dependent on how this reward function is defined. For example, if the reward function prioritized staying *on-script* then it would strive to return to a state where future events are predicted with high probability.

There are many advantages to this type of approach. First, it does not rely on external ontologies to build up a representation of state. These neural models can be trained on corpora of natural language, such as stories or news articles, including non-English corpora. In addition, this allows the system to easily learn different types of behavior based on the corpus used for training. The previously proposed architecture uses plot graphs to encode commonsense procedures and then provides strategies for handling unexpected user behavior. What a neural net *expects* would depend on the data it is trained upon; for example, training it on plot synopses of movies would naturally lead to expectations of dramatic behavior from the user and more commonsense behavior would be considered exceptional. One disadvantage of a neural network approach is that the state representation used by the RNN is obfuscated in the hidden LSTM layers. Thus, it is not clear as to why the system will make certain choices (beyond the goal of maximizing future reward). This loss in system transparency can make it difficult to evaluate the effectiveness of such a system. Since state cannot be directly observed, this leads to a greater likelihood of non-sequiturs due to mistaken beliefs about the state of the fictional improv world.

6 Conclusions

In this paper, we introduce *improvisational storytelling*, one or more people constructing a story in real time without advanced notice of topic or theme. We discuss some of the challenges that need to be addressed in order to create a

computational, improvisational storytelling system and propose two architectures that address some of these challenges as a starting point.

As human-AI interaction becomes more common, it becomes more important for AIs to be able to engage in open-world improvisational storytelling. This is because it enables AIs to communicate with humans in a natural way without sacrificing the human's perception of agency. We hope that formalizing the problem and examining the challenges associated with improvisational storytelling will encourage researchers to explore this important area of work to help enable a future where AI systems and humans can seamlessly communicate with one another.

Acknowledgements. This work was supported by the Defense Advanced Research Projects Agency (DARPA) under Contract No. W911NF-15-C-0246. The authors would also like to thank Will Hancock for his work on our initial plot graph system.

References

1. Riedl MO, Bulitko V (2013) Interactive Narrative: An Intelligent Systems Approach. *AI Mag* 34:67–77. doi: 10.1609/aimag.v34i1.2449
2. Bates J (1992) Virtual Reality, Art, and Entertainment. *Presence Teleoperators Virtual Environ* 1:133–138. doi: 10.1162/pres.1992.1.1.133
3. Riedl MO, Stern A, Dini DM, Alderman JM (2008) Dynamic Experience Management in Virtual Worlds for Entertainment, Education, and Training. *Int Trans Syst Sci Appl Spec Issue Agent Based Syst Hum Learn* 4:23–42.
4. Riedl M, Saretto CJ, Young RM (2003) Managing Interaction Between Users and Agents in a Multi-agent Storytelling Environment. In: *Proc. Second Int. Jt. Conf. Auton. Agents Multiagent Syst.* ACM, pp 741–748
5. Porteous J, Cavazza M (2009) Controlling narrative generation with planning trajectories: The role of constraints. In: *Jt. Int. Conf. Interact. Digit. Storytell.* Springer, pp 234–245
6. Nelson MJ, Mateas M (2005) Search-Based Drama Management in the Interactive Fiction Anchorhead. In: *Proc. First Annu. Conf. Artif. Intell. Interact. Digit. Entertain.* pp 99–104
7. Weyhrauch PW (1997) *Guiding Interactive Drama.* Carnegie Mellon University, Pittsburgh, PA
8. Nelson MJ, Roberts DL, Isbell CL, Mateas M (2006) Reinforcement learning for declarative optimization-based drama management. In: *Proc. Fifth Int. Jt. Conf. Auton. Agents Multiagent Syst.* ACM, pp 775–782
9. Diaz-Agudo B, Gervas P, Peinado F (2004) A case based reasoning approach to story plot generation. In: *Adv. Case-Based Reason. Proc.* Springer, pp 142–156
10. Zhu J, Ontañón S (2014) Shall I compare thee to another story?—An empirical study of analogy-based story generation. *IEEE Trans Comput Intell AI Games* 6:216–227. doi: 10.1109/TCIAIG.2013.2275165
11. Ware SG, Young RM (2010) Rethinking Traditional Planning Assumptions to Facilitate Narrative Generation. *Proc 2nd Work Comput Model Narrat (CMN)*

- 2010) 71–72.
12. Mateas M, Stern A (2003) Integrating Plot, Character and Natural Language Processing in the Interactive Drama Façade. In: Proc. Technol. Interact. Digit. Storytell. Entertain. Conf. pp 139–151
 13. Guzdial M, Harrison B, Li B, Riedl MO (2015) Crowdsourcing Open Interactive Narrative. Proc. 10th Int. Conf. Found. Digit. Games
 14. Swanson R, Gordon AS (2008) Say Anything: A Massively Collaborative Open Domain Story Writing Companion. In: Jt. Int. Conf. Interact. Digit. Storytell. Springer, pp 32–40
 15. Magerko B, Manzoul W, Riedl M, et al (2009) An Empirical Study of Cognition and Theatrical Improvisation. In: Proc. Seventh ACM Conf. Creat. Cogn. ACM, pp 117–126
 16. Magerko B, O’Neill B (2012) Formal Models of Western Films for Interactive Narrative Technologies. In: Proc. 2nd Work. Comput. Model. Narrat. Istanbul, Turkey, pp 83–90
 17. Brisson A, Magerko B, Paiva A (2011) Tilt riders: Improvisational agents who know what the scene is about. In: Int. Work. Intell. Virtual Agents. Springer, pp 35–41
 18. Li B, Lee-Urban S, Johnston G, Riedl MO (2013) Story Generation with Crowdsourced Plot Graphs. In: Proc. Twenty-Seventh AAAI Conf. Artif. Intell. pp 598–604
 19. Chambers N, Jurafsky D (2008) Unsupervised Learning of Narrative Event Chains. In: Proc. Assoc. Comput. Linguist. Citeseer, pp 789–797
 20. Pichotta K, Mooney RJ (2016) Learning Statistical Scripts With LSTM Recurrent Neural Networks. Proc. 30th AAAI Conf. Artif. Intell. (AAAI-16), Phoenix, Arizona
 21. Kipper-Schuler K (2005) VerbNet: A broad-coverage, comprehensive verb lexicon. University of Pennsylvania
 22. Zwaan RA, Langston MC, Graesser AC (1995) The Construction of Situation Models in Narrative Comprehension: An Event-Indexing Model. *Psychol Sci* 6:292–297. doi: 10.1111/j.1467-9280.1995.tb00513.x
 23. Niehaus J, Michael Young R (2014) Cognitive models of discourse comprehension for narrative generation. *Lit Linguist Comput* 29:561–582.
 24. O’Neill B, Riedl M (2014) Dramatis: A Computational Model of Suspense. *AAAI* 2:944–950.
 25. Riedl M, Li B, Ai H, Ram A (2011) Robust and Authorable Multiplayer Storytelling Experiences. In: Proc. Seventh AAAI Conf. Artif. Intell. Interact. Digit. Entertain. pp 189–194
 26. Speer R, Havasi C (2012) Representing General Relational Knowledge in ConceptNet 5. In: LREC. pp 3679–3686
 27. Branavan SRK, Kushman N, Lei T, Barzilay R (2012) Learning High-Level Planning from Text. In: Proc. 50th Annu. Meet. Assoc. Comput. Linguist. pp 126–135
 28. Li J, Monroe W, Ritter A, Jurafsky D (2016) Deep Reinforcement Learning for Dialogue Generation. arXiv Prepr. arXiv1606.01541