Improvisational Storytelling Agents

Lara J. Martin, Prithviraj Ammanabrolu, Xinyu Wang, Shruti Singh, Brent Harrison, Murtaza Dhuliawala, Pradyumna Tambwekar, Animesh Mehta, Richa Arora, Nathan Dass, Chris Purdy, and Mark O. Riedl
Georgia Institute of Technology

Abstract

The problem of improvisational story generation involves one or more agents collaborating in order to create a story without any advance notice of topic. We present a pipeline for an artificial agent that is capable of improvisational storytelling while collaborating with a human agent. Starting with story corpora, we "eventify" sentences, which creates a simplified and abstracted representation. The rest of the pipeline–the agent's response–is broken into three parts: generating successive events (event-to-event), translating of events back into natural language (event-to-sentence), and plugging the specifics of the story back into the generated sentences (slot filling). We discuss techniques for each of these sub-problems.

1 Introduction

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, in team planning and brainstorming, or in table-top role-playing games. We propose a grand challenge of creating artificial agents capable of engaging with humans in improvisational, real-time storytelling. Aside from being a testbed problem domain for computational creativity, improvisational storytelling addresses several challenges found in many other AI and ML domains: long-term context understanding, commonsense knowledge, dialogue planning, and humor and metaphor understanding. Improvisational storytelling is closely related to automated story generation [1] except for the presence of a human co-creator that can change the direction and context of the story at any time.

Improvisational storytelling is a computationally challenging problem. The space of possible stories that can be created is very large. 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. 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. For more about the grand challenge, see [2].

2 A Framework for Improvisational Storytelling Agents

We propose a framework for intelligent agents that learn to improvise from large story corpora (see Figure 1).

2.1 Eventification

The dimensionality of the story search space derived from a story corpus is very large and sparse. Due to language variation, we are unlikely to see the same pair of sentences more than once in an entire

31st Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA.

corpus. In our framework, we reduce the dimensionality of the space of stories by taking each sentence and transforming it into an event. We define an *event* as the tuple $\langle s, v, o, m \rangle$ where v is a verb, s is the subject of the verb, o is the object of the verb, and m is a modifier that provides additional context, such as an indirect object, propositional object, causal complement, or any unknown dependency. Martin et al. [3] show that replacing character names with generic placeholders and otherwise using WordNet [4] synsets for s, o, and m, and using VerbNet [5] frames for v works best.

By reducing sentences to events, we reduce the sparsity of the story space—we are more likely to see the same event more than once. Human participant sentences are also reduced to events, making it more likely that human input matches events the agent has experienced during learning.

2.2 Event-to-Event

The Event-to-Event module takes in one or more events from the human participant and generates what it believes to be the next event(s). There are a number of techniques that can be used to generate story continuations. Recurrent neural networks such as sequence-to-sequence networks [6], using long-short term memory (LSTM) [7] cells, treat story generation as sampling from a specialized language model distribution trained on an eventified story corpus. Given one or more events, it generates one or more events most likely to follow the current history of events.

Storytelling, like dialogue generation, is a goal-driven process. Although the human participant may alter the context unpredictably, an agent may want to pick the event(s) that make the most sense given the current context but also afford future storytelling opportunities. Sampling from a language model is not goal-driven. One alternative is to use text-based deep reinforcement learning. A deep reinforcement learner can learn a new distribution based on a reward signal that encodes good storytelling practices. See [8] for one potential technique. Another alternative, Markov-chain Monte Carlo [9], can also learn to alter the language model to increase the likelihood of generating certain types of events.

2.3 Event-to-Sentence

Events are not easily human readable, especially when verbs are VerbNet frames and subjects and objects are WordNet synsets. Once we have generated a new successor event, we must produce a human-readable natural language sentence. This is a machine translation problem; a sequence-to-sequence neural network [6] is used. We find that a beam search decoding strategy produces more natural-sounding sentences.

Generalized events reduce the chance of the Event-to-Event generator becoming confused by specific names or nouns. Event-to-Sentence is trained to translate generalized events to generalized sentences which have their named entities replaced by placeholder symbols and other nouns replaced by WordNet synsets. Verbs are not generalized in these sentences since there are fewer verbs per VerbNet category. This also enables the Event-to-Sentence network to learn the correct English grammar with the verb in place. Generalizing decreases translation error by pushing the grounding of specific names and nouns to a later stage.

2.4 Memory and Slot Filling

Semantic slot filling is a common practice in dialog systems research that maintains coherence and state by extracting specifics and fitting them into categories (e.g. [10]). Our agent framework implements a long-term memory based on *event indexing* from psychology [11]. Each event is stored as a node in a graph, and each entity is also a node that is associated with one or more event nodes. The named entities and nouns in generalized sentences produced by Event-to-Sentence can be filled back in by searching the memory graph for the most salient entities. It is possible for Event-to-Event to generate more entity placeholders than have previously been used. In these cases, slot filling is handled by inventing new names by pulling them from the US Census or picking new nouns from WordNet hierarchies.

Table 1 shows example sentences generated using different implementations of Event-to-Event: a sequence-to-sequence neural network, a deep reinforcement learner, and Markov-chain Monte Carlo search.

Acknowledgments

This work was supported by the Defense Advanced Research Projects Agency (DARPA) under Contract No. W911NF-15-C-0246.

References

[1] R. M. Young, S. Ware, B. Cassell, and J. Robertson, “Plans and planning in narrative generation: a review of plan-based approaches to the generation of story, discourse and interactivity in narratives,” *Sprache und Datenverarbeitung, Special Issue on Formal and Computational Models of Narrative*, vol. 37, no. 1-2, pp. 41–64, 2013.

[2] L. J. Martin, B. Harrison, and M. O. Riedl, *Improvisational Computational Storytelling in Open Worlds*, pp. 73–84. Springer International Publishing, 2016.

[3] L. J. Martin, P. Ammanabrolu, X. Wang, W. Hancock, S. Singh, B. Harrison, and M. O. Riedl, “Event representations for automated story generation with deep neural nets.” Submitted for publication, 2018.

[4] C. Fellbaum, *WordNet*. Wiley Online Library, 1998.

[5] K. K. Schuler, “Verbnet: A broad-coverage, comprehensive verb lexicon,” 2005.

[6] I. Sutskever, O. Vinyals, and Q. V. Le, “Sequence to sequence learning with neural networks,” in *Advances in neural information processing systems*, pp. 3104–3112, 2014.

[7] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[8] P. Tambwekar, M. Dhuliawala, A. Mehta, N. Dass, B. Harrison, and M. O. Riedl, “Interactive story generation using deep reinforcement learning.” Submitted for publication, 2017.

[9] B. Harrison, C. Purdy, and M. O. Riedl, “Toward automated story generation with markov chain monte carlo methods and deep neural networks,” in *Proceedings of the 2017 Workshop on Intelligent Narrative Technologies*, 2017.

[10] Y.-N. Chen, W. Y. Wang, and A. I. Rudnicky, “Unsupervised induction and filling of semantic slots for spoken dialogue systems using frame-semantic parsing,” in *Automatic Speech Recognition and Understanding (ASRU), 2013 IEEE Workshop on*, pp. 120–125, IEEE, 2013.

[11] R. A. Zwaan, M. C. Langston, and A. C. Graesser, “The construction of situation models in narrative comprehension: An event-indexing model,” *Psychological science*, vol. 6, no. 5, pp. 292–297, 1995.

Appendix

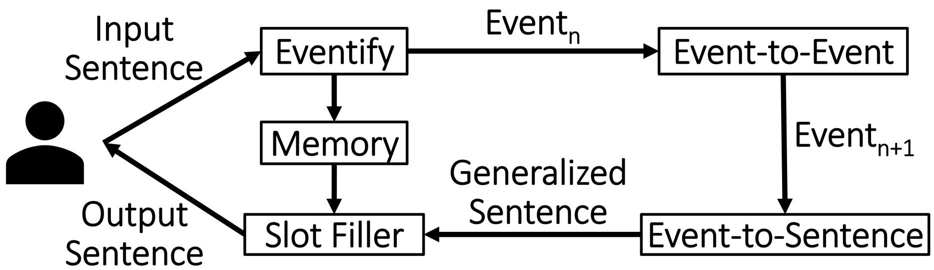

Figure 1: Our automated story generation pipeline.

Table 1: Inputs and final output pairs from the different versions of our pipeline.

Experiment	**Input (Previously Unseen)**	**Generated Next Sentence(s), Filled In**
Sequence-to-Sequence LSTM	Caleb hated the evil wizard.	Sue meets the wizard.
Deep Reinforcement Learner (DRL)	John rode his horse to the bank.	John theseus the horse off the bank. The horse proves himself as a lawn chair entrepreneur. The give phase the reassigns cat. Oilman possesses John Kelley s lover Loneta Denise. Together bank slices the kidnaps uniform. John is confronts to owes the main bank of the improvised snuffler in the stain. The phony faller then arrives back to the that. The that marry a great anger in hate.
Markov-chain Monte Carlo (MCMC)	Jon and Mary walk to the bridge.	Enrolled Jon walks into the bridge. Jon is gathers by she yellow man. The financial institution threatens Jon volume parent in respects. Willing to starting. Jon sneaks out. Sorus is saved by a criminal. The criminal accidentally gets that warplane failure. Scattered is being held at the mortie piece length. He agrees to marry she. What is shared. Jon is forced to live.