Story Generation via Deep Reinforcement Learning

Pradyumna Tambwekar, Murtaza Dhuliawala, Animesh Mehta, Nathan Dass, Brent Harrison, and Mark O. Riedl
Georgia Institute of Technology

Abstract

Automated story generation is the problem of automatically selecting a sequence of events, actions, or words that can be told as a story. We introduce a deep reinforcement learning approach to story generation trained on a textual story corpus. Unlike other neural network based approaches to story generation, a reward function allows a human user to control the direction that the story follows.

1 Introduction

Automated story generation is the problem of automatically selecting a sequence of events, actions, or words that can be told as a story. In this paper, we introduce a deep reinforcement learning approach to automated story generation in which the generator learns how to tell stories from large text-based corpora such as book and movie plot summaries mined from Wikipedia.

One way to teach a system how to tell stories with a corpus is to use Recurrent Neural Networks (RNNs), particularly Long Short-Term Memory (LSTM) networks [1]. LSTM networks have recently been trained on story corpora in order to generate novel stories [2–4]. LSTMs can be thought of as learning a distribution over successor tokens—characters, words, or sentences. Thus, story generation with RNNs is a process of iteratively sampling from the learned distribution. However, this presents two challenges. First, LSTMs as language models don't have any incentive to maintain context. Any sequence of more than a few sentences will tend to lose narrative coherence. Second, LSTMs are not goal-driven. It is unlikely that they will produce a story with a given ending or structure.

Reinforcement learning algorithms use trial-and-error search to find a policy that achieves the highest expected reward. In the case of story generation, we reward the system for generating stories that meet a desired authorial goal while also creating stories that look like those from the training corpus. For example, we might reward the generator for generating a story in which characters meet, fall in love, and marry. Our particular deep reinforcement learning approach uses policy gradients to iteratively train a neural network, so that the distribution it learns over successor events increases the likelihood that the target events that occur follow the direction dictated by the user.

2 Background

Story creation is a planning process—it is driven by the author's goals as a storyteller[5]. To date, most story generation systems have used symbolic planning algorithms [6–9] or case-based planning [10]in well-defined micro-worlds. Other approaches learn from crowdsourced corpora [11] or retrieve sentences of stories from blogs [12]. Language modeling neural network approaches to story generation include [2–4]. Reinforcement learning has been used to generate interactive narrative, where a player assumes the role of a character in a virtual world and can influence the state of the world through his or her actions [13, 14]. Harrison et al. [15] uses Markov Chain Monte Carlo search trained on stories.

Our work follows [3] in reducing the dimensionality of a training corpus of stories by transforming sentences into *events*, where an event is a tuple $\langle s, v, o, m \rangle$ such that s is the subject of the verb, v is

a verb, o is the direct object, and m is a modifier that provides additional context, such as an indirect object, propositional object, causal complement, or unknown dependency. Martin et al. [3] show that replacing named entities with generic placeholders and otherwise using WordNet [16] synsets for s, o, and m, and using VerbNet [17] frames for v further improves generation. They also provide a technique for translating events back into natural language. Events further allow interactivity. A human user can change the state of the story by introducing a new sentence; by translating this sentence into an event, it is more likely to correlate to known events that the system has been trained on.

Our technique shares similarities with policy-gradient reinforcement learning approaches to dialogue generation [18]. Whereas reinforcement for dialogue generation uses the reward function to correct for consistency, coherence, and repetition during neural net decoding, our approach incorporates external authorial goals into the reward function.

3 Deep Reinforcement Learner

Our technique starts with a sequence-to-sequence network [19] with LSTM cells that was pre-trained on a story corpus. This model approximates the distribution over successor events given a predecessor event. Thus, the set of actions in our domain space is the set of all possible events, with no legality constraints on what sentences can follow each other. Because of the size of our search space, we use a variation of policy-gradient search [20]. See Figure 1 for a diagrammatic illustration of our training process.

The training process can be thought of as iteratively shifting the distribution over successor events from the pre-trained network to one that increases the likelihood of generating rewarding events. In our experiments, we reward the system for generating stories that contain a given target verb, such as "meet", "fall in love" ('glossed as 'admire" in VerbNet), or "marry". This gives a human user overarching control over the direction of the stories that are generated.

To train the policy network, we iteratively choose events from the training corpus and sample from the distribution of successor events approximated by the network. We calculate a reward (described below) and the reward is multiplied with the probability of the particular event actually succeeding its predecessor. This value is backpropagated into the policy network.

Rewards are sparse in our domain and our technique will only learn if it sees rewarding events often. Early in the training phase, we periodically replace the expected output with a target event (which meets the author's intent) so that it yields a positive reward. This has the effect of rapidly shifting the model early and allowing it to slowly revert back, thus achieving a model that balances both following the original pre-trained distribution, and driving toward a target event. We anneal the frequency of the positive reward (being artificially introduced) as training proceeds.

Reward is calculated as follows. Unlike prior policy gradient methods that use the likelihood of outputs sampled from an RNN, we estimate the Bayesian probabilities from our training data and utilize these probabilities. The objective of our system is thus to maximize

$$\frac{1}{n}\sum_{i} R_i \frac{p(y_i|X_i, X_{i-1}, X_{i-2}) \prod_j p(y_{i,k}|y_{i,j})}{p(y_i|X_i, X_{i-1}, X_{i-2}) + \prod_j p(y_{i,k}|y_{i,j})} \tag{1}$$

where n is batch size (10 in our experiments) and where each event y_i gives us an authorial reward of R_i. X_i, X_{i-1}, X_{i-2} denote the three preceding events of the story before the sample y_i and $\prod_j p(y_{i,k}|y_{i,j})$ estimates the probability of the k^{th} token in the sample given the previous j tokens in the same event. These probabilities together encapsulate both inter- and intra-event coherence. The use of Bayesian probabilities affords us the ability to estimate the probability of the event with respect to the story corpus instead of just the previous event. In cases where the reward falls short of a predefined threshold, the model calculates the loss with respect to the output of a pre-trained sequence-to-sequence model.

Table 3 shows sample table output using only romance movie plots from the CMU Movie Plot Summary corpus [21]. Events generated by the reinforcement learner are converted to natural language using the technique described in [3]. Story coherence (and grammar) is expected to improve with a larger training corpus; we note that in all cases the reinforcement learner succeeds in generating target events in a non-trivial manner.

Acknowledgments

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

References

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

[2] M. Roemmele, S. Kobayashi, N. Inoue, and A. M. Gordon, "An rnn-based binary classifier for the story cloze test," *LSDSem 2017*, p. 74, 2017.

[3] L. J. Martin, P. Ammanabrolu, W. Hancock, S. Singh, B. Harrison, and M. O. Riedl, "Event representations for automated story generation with deep neural nets," *CoRR*, vol. abs/1706.01331, 2017.

[4] A. Khalifa, G. A. Barros, and J. Togelius, "Deeptingle," *arXiv preprint arXiv:1705.03557*, 2017.

[5] M. Sharples, *How We Write: Writing as Creative Design*. London: Routledge, 1999.

[6] J. R. Meehan, "TALE-SPIN: An interactive program that writes stories," in *Proceedings of the 5th International Joint Conference on Artificial Intelligence*, pp. 91–98, 1977.

[7] M. Lebowitz, "Planning stories," in *Proceedings of the 9th Annual Conference of the Cognitive Science Society*, pp. 234–242, 1987.

[8] R. Pérez y Pérez and M. Sharples, "MEXICA: A computer model of a cognitive account of creative writing," *Journal of Experimental and Theoretical Artificial Intelligence*, vol. 13, pp. 119–139, 2001.

[9] and R. Michael Young, "Narrative planning: Balancing plot and character," *Journal of Artificial Intelligence Research*, vol. 39, pp. 217–268, 2010.

[10] P. Gervás, B. Díaz-Agudo, F. Peinado, and R. Hervás, "Story plot generation based on cbr," *Knowledge-Based Systems*, vol. 18, no. 4, pp. 235–242, 2005.

[11] B. Li, S. Lee-Urban, G. Johnston, and M. O. Riedl, "Story generation with crowdsourced plot graphs," in *Proceedings of the 27th AAAI Conference on Artificial Intelligence*, (Bellevue, Washington), July 2013.

[12] R. Swanson and A. Gordon, "Say Anything: Using textual case-based reasoning to enable open-domain interactive storytelling," *ACM Transactions on Interactive Intelligent Systems*, vol. 2, no. 3, pp. 16:1–16:35, 2012.

[13] P. Wang, J. Rowe, W. Min, B. Mott, and J. Lester, "Interactive narrative personalization with deep reinforcement learning," *The Twenty-Sixth International Joint Conference on Artificial Intelligence*, vol. 3852-3858.

[14] D. L. Roberts, M. Nelson, C. Isbell, M. Mateas, and M. Littman, "Targeting specific distributions of trajectories in MDPs," in *Proceedings of the 21st National Conference on Artificial Intelligence*, 2006.

[15] 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 AAAI Workshop on Intelligent Narrative Technologies*, 2017.

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

[17] K. K. Schuler, "Verbnet: A broad-coverage, comprehensive verb lexicon," 2005.

[18] J. Li, W. Monroe, A. Ritter, M. Galley, J. Gao, and D. Jurafsky, "Deep reinforcement learning for dialogue generation," *arXiv:1606.01541*, 29 Sep 2016.

[19] 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.

[20] R. S. Sutton, D. McAllester, S. Singh, and Y. Mansour, "Policy gradient methods for reinforcement learning with function approximation," *Advances in Neural Information Processing Systems*, vol. 12, p. 1057–1063, 2000.

[21] D. Bamman, B. O'Connor, and N. A. Smith, "Learning latent personas of film characters," in *Proceedings ACL2013*, 2013.

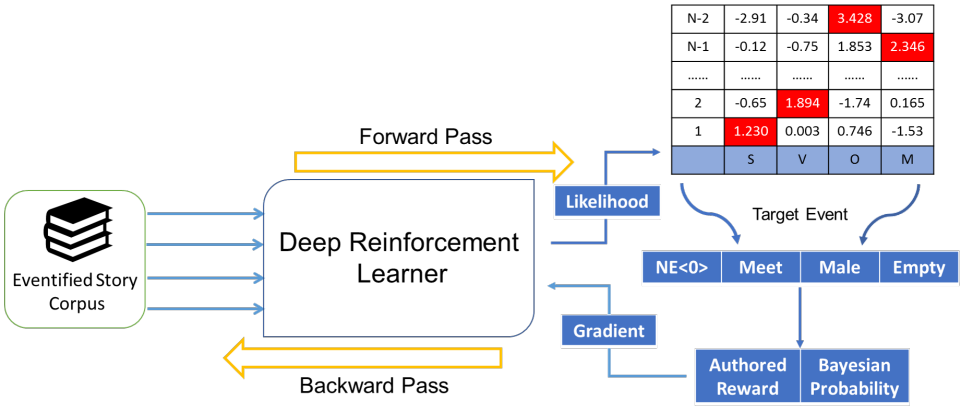

Figure 1: Deep Reinforcement learning pipeline. Corpus containing sequential events from stories is used to train the model. The model generates a likelihood over the vocabulary words. A target event is sampled. Reward is computed on this with respect to the input event and combined with loss with respect to the target event. Gradient is computed on this loss and backpropagated.

Table 1: Input and Story generated using our system. Each row shows the model generating events, which leads to an event containing the target verb and ends on that sentence. We have the sentences translated from the event representations along with the corresponding event representation shown here.

Input	**Target Verb**	**Raw Events (Above)/Generated sentences (Below)**
Jack and Jill went up the hill.	Meet	1. <NE0> meander-47.7 EmptyModifier Synset('geological-formation.n.01) 2. <NE>1 meet-36.3-1 meander-47.7 EmptyModifier
		Jack and Arlyn go to the hill.That time Jack and Arretta meet up with the Remorseful.
That time Jack and Arretta meet up with the Remorseful.	Admire	1. EmptyObject EmptyObject EmptyObject <NE>1 2. <NE>0 admire-31.2 EmptyObject Synset('subjugation.n.01')
		One duration actual and Jack cries as Nachman Elana Whom Jack believes to be a surrender.
Jack believes to be a surrender.	Marry	1. Synset('person.n.01') forbid-67-1 EmptyObject Synset('genitor.n.01') 2. <NE>0 marry-36.2 Synset('odd-toed-ungulate.n.01') EmptyModifier
		The two I are unable to stop Jack from his parent. Jack and Dimple marry a rhinoceros in the pianism.

Table 2: Input and Story generated using our system. Each row shows the model generating events which leads to an event containing the target verb and ends on that sentence. The last column shows the raw events in each case and the natural language translations for those events below.

Input	**Target Verb**	**Generated sentences**
John rode his horse to the bank.	Meet	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.
Together bank slices the kidnaps uniform.	Admire	John is confronts to owes the main bank of the improvised snuffler in the stain.
John is confronts to owes the main bank of the improvised snuffler in the stain.	Marry	The phony faller then arrives back to the that. The that marry a great anger in hate.

Table 3: Input and Story generated using our system. Each row shows the model generating events, which leads to a story containing the target verb and ends on that sentence. We have the sentences translated from the event representations shown here.

Input	**Target Verb**	**Generated sentences**
Ryan walked to the beach with his guitar.	Meet	The craftsman and Ryan Monte are traveling to the secure beach.The critic will be able to quit his own rings beverly out.Ryan and Wade born metempsychosis the city s pickup on space.Ryan klutzy serious the admiring voice.Ryan and Fernando decide to be set.Where she meets she with Plotting.
she meets she with Plotting.	Admire	When he catches murky the acquaintance.Ryan turns to display the euthanasia.
Ryan turns to display the euthanasia.	Marry	The that wins Ryan with Ryan.When Ryan became directed concentration marital status to Antone.