Story Generation with Deep Reinforcement Learning

Georgia Institute of Technology | Entertainment Intelligence Lab | College of Computing
Nathan Dass, Murtaza Dhuliwala, Animesh Mehta, Lara Martin, Raj Ammanabrolu, Dr. Mark Riedl

entertainment intelligence lab

Introduction

Story generation is the problem of automatically selecting a sequence of events that can be told as a story. Story generation is knowledge-intensive; traditional story generators rely on prior knowledge about fictional worlds, including characters, places, and actions that can be performed.

Story-generating systems can be considered similar to creating a conversational or chatbot systems, with a few additional factors to take into account:

- Causality - is everything that has happened taken into account for future events in the story?
- Narrative arcs - the overall flow of events in the story
- Creativity - surprising, valuable, and novel

Previous attempts find a solution:

- Deep reinforcement relevant networks [1]
- Dynamic memory networks [2]
- Adversarial learning [3]

We use deep reinforcement learning to combine learning from raw input with learning by rewards and punishments to solve the story-generation problem. With deep reinforcement learning, we can often obtain performance comparable to humans for very complex problems.

Sentence to Event

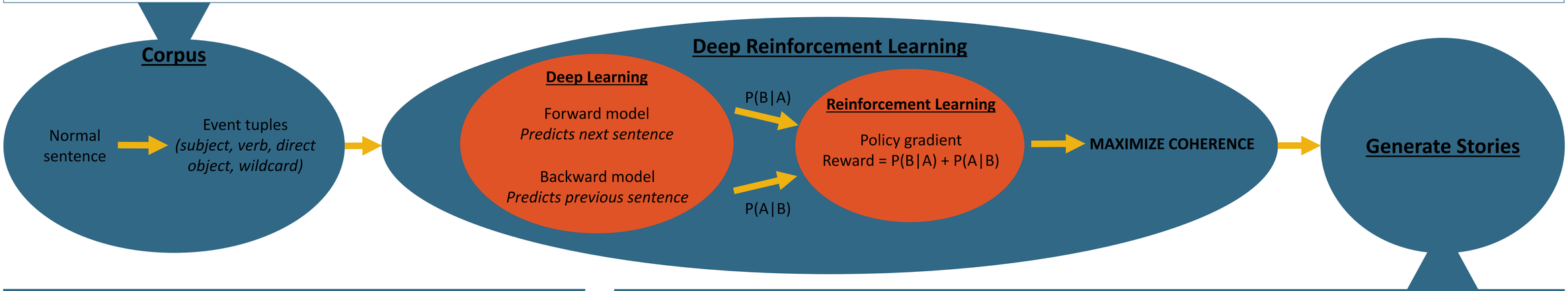
- Extracts semantic information from a sentence using dependency parsing and named-entity recognition
- Format - (subject, verb, direct object, wildcard)
- Abstracted using VerbNet & WordNet to find categories
- Event2Event - finds the best event structure
  - For each type, we use a sequence-to-sequence LSTM RNN neural network to generate the next event in the story
  - Score similarity of generated next event and actual next event

"Ranchhod Rai Patel is an egotistical and arrogant self made Gujarati man who arrived from India and settled in the New York area years ago"

Original words:
(man, arrive, EmptyParameter, settle)

Abstracted words:
(male, contiguous_location, EmptyParameter, travel)

	Perplexity	BLEU
Original Sentence	872.63	0.039
Specific	748.91	0.188
Generic	54.23	0.057



Event to Sentence

Stacks

- Specifics are stored in stacks according to their general category
- When an "abstracted" or "generic" event is seen, pop from the stack
- If empty, look up a new word in that category on WordNet/VerbNet

Translation

- The filled-in (specific) event is put into another sequence-to-sequence LSTM RNN to "translate" it back into English
- Trained on the same data from generating the events

Story Generation

The policy gradient model is built using a sequence-to-sequence model, which leverages two models to generate rewards.

- Forward model - predicts the next event given the current event; generates stories in order
- Backward model - predicts the previous event given the current event; generate stories in reverse order

<start_of_story>
hard,assassins,believe,Synset('group.n.01')
Synset('person.n.01'),send,Synset('written_communication.n.01'),<NE>0
<NE>0,appall,Synset('entity.n.01'),Synset('act.n.02')
Synset('person.n.01'),give,EmptyParameter,Synset('social_group.n.01')
<NE>0,offers,return,which
Synset('male.n.02'),accept,<NE>0,Synset('event.n.01')
<end_of_story>

References

1. He, Ji, et al. "Deep reinforcement learning with a natural language action space." arXiv preprint arXiv:1511.04636 (2015).
2. Kumar, Ankit, et al. "Ask me anything: Dynamic memory networks for natural language processing." CoRR, abs/1506.07285 (2015).
3. Li, Jiwei, et al. "Adversarial Learning for Neural Dialogue Generation." arXiv preprint arXiv:1701.06547 (2017).
4. Li, Jiwei, et al. "Deep reinforcement learning for dialogue generation." arXiv preprint arXiv:1606.01541 (2016).