Story Generation

Plan, Write, and Revise: an Interactive System for Open-Domain Story Generation

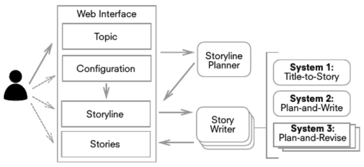


Figure 1: Diagram of human-computer interaction mediated by the the demo system. The dotted arrows represent optional interactions that the user can take. Depending on the set-up, the user may choose to interact with one or all story models.

tem can address specific shortcomings of generated stories. We identify three types of weaknesses common to generation systems – *Creativity*, *Relevance*, and *Causal & Temporal Coherence*, and conduct experiments where the human is instructed to focus on improving specifically one of them. The targeted human improvement areas intentionally match the *Plan-and-Revise* discriminators, so that, if successful, the ”human discriminator” data can assist in training the machine discriminators. All experiments (save experiment 2, which lets the user pick between models) use the *Plan-and-Revise* system.

Strategies for Structuring Story Generation

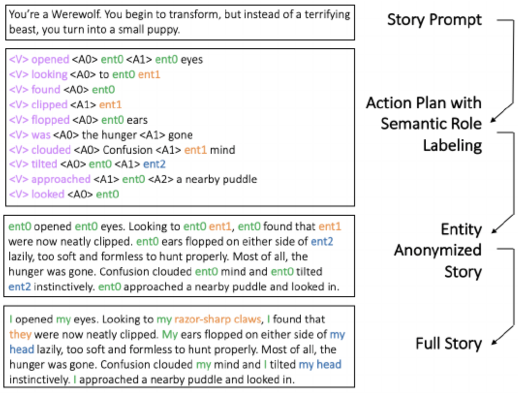


Figure 1: Proposed Model. Conditioned upon the prompt, we generate sequences of predicates and arguments. Then, a story is generated with placeholder entities such as *ent0*. Finally we replace the placeholders with specific references.

A Knowledge-Enhanced Pretraining Model for Commonsense Story Generation

Context:
The first time I saw the results of an **accident**, I was in a bus.

Golden Continuation by Human:
We were stopped and a friend pointed out the accident on the road. A man was under a car after having been run over. People tried to move the car off him but only made it worse. After seeing that, I am much more careful when crossing the street.

pretrained GPT-2:
She was trying to get home but the other passengers would not let her go. He thought she was going to die because of her weight. *She was trying to get home but the other passengers would not let her go.* The bus driver *would not let her go.*

Fine-tuned GPT-2:
I was on my way to a party. I was on my way to a party and I 'd gotten out of my seat, and started driving. I got a flat tire, so I stopped driving. I drove to the party and had a great time.

Table 1: Story examples generated by human and GPT-2 models. The stories written by the pretrained GPT-2 and fine-tuned GPT-2 (post-trained on ROCStories (Mostafazadeh et al., 2016b)) suffer from repetition (in *italic*), bad inter-sentence coherence to the context (e.g., ignoring key entities such as `accident` in **bold**), as well as conflicting logic (underlined, e.g., first `stopped driving` and then `drove to the party`), in spite of their good fluency and intra-sentence coherence.

We propose a knowledge-enhanced pretraining model for commonsense story generation by extending GPT-2 with external commonsense knowledge. The model is post-trained on the knowledge examples constructed from ConceptNet and ATOMIC, thereby improving long-range coherence of generated stories.

To generate reasonable stories, we adopt a classification task to distinguish true stories from auto-constructed fake stories. The auxiliary task makes the model implicitly capture the causal, temporal dependencies between sentences and inter-sentence coherence, and lead to less repetition.

We conduct extensive experiments with automatic and manual evaluation. Results show that our model can generate more reasonable stories than strong baselines, particularly in terms of logicality and global coherence.[1]

Event Representations for Automated Story Generation with Deep Neural Nets

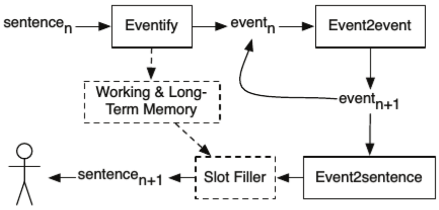


Figure 1: Our automated story generation pipeline. Dashed boxes and arrows represent future work.

Event2Sentence

Unfortunately, events are not human-readable and must be converted to natural language sentences. Since the conver-

sion from sentences to (multiple) events for *event2event* is a linear and lossy process, the translation of events back to sentences is non-trivial as it requires adding details back in. For example, the event ⟨relative.n.01, characterize-29.2, male.n.02, feeling.n.01⟩ could, hypothetically, have come from the sentence "Her brother praised the boy for his empathy." In actuality, this event came from the sentence "His uncle however regards him with disgust."

Story Realization: Expanding Plot Events into Sentences

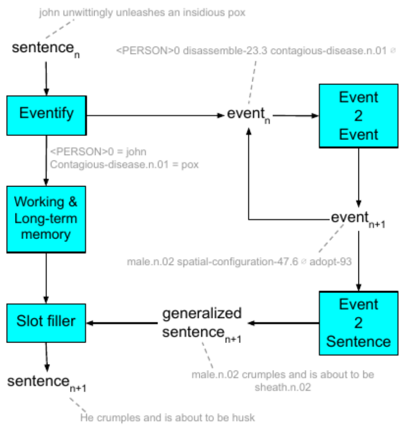


Figure 1: The full automated story generation pipeline, illustrating an example where the event-to-event module generates only a single following event.

Learning to Predict Explainable Plots for Neural Story Generation

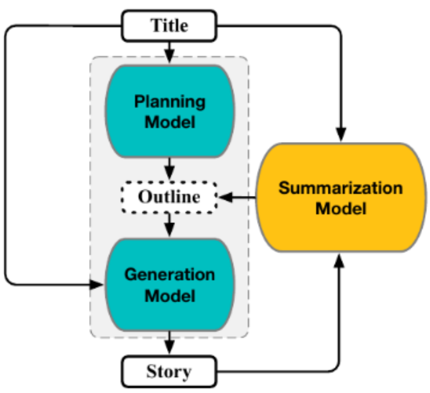


Figure 2: An overview of our latent variable model

for NSG. Note that the summarization model is only used in the training phase while the planning model and the generation model are involved in both the training and inference phases.

1. DIRECT (Roemmele, 2016): Directly generating a story given a title using sequence-to-sequence models.
2. SKELETON (Xu et al., 2018): First generating a skeleton and then expanding the skeleton into a complete sentence until the whole story has been generated.
3. HIERARCHICAL (Fan et al., 2018): First generating a prompt and then generating a story based on the prompt.
4. SEPARATE (Drissi et al., 2018): First generating an outline and then generating a story based on the outline using two separate modules.
5. PLANNED (Yao et al., 2019): First generating a storyline and then generating a story based on the title and the storyline.

Learning Norms from Stories: A Prior for Value Aligned Agents

Value alignment is a property of an intelligent agent indicating that it can only pursue goals and activities that are beneficial to humans. Traditional approaches to value alignment use imitation learning or preference learning to infer the values of humans by observing their behavior. We introduce a complementary technique in which a value-aligned prior is learned from naturally occurring stories which encode societal norms. Training data is sourced from the children's educational comic strip, *Goofus & Gallant*. In this work, we train multiple machine learning models to classify natural language descriptions of situations found in the comic strip as normative or non-normative by identifying if they align with the main characters' behavior. We also report the models' performance when transferring to two unrelated tasks with little to no additional training on the new task.

Fine-Tuning a Transformer-Based Language Model to Avoid Generating Non-Normative Text

Large-scale, transformer-based language models such as GPT-2 are pretrained on diverse corpora scraped from the internet. Consequently, they are prone to generating content that one might find inappropriate or non-normative (i.e. in violation of social norms). In this paper, we describe a technique for fine-tuning GPT-2 such that the amount of non-normative content generated is significantly reduced. A model capable of classifying normative behavior is used to produce an additional reward signal; a policy gradient reinforcement learning technique uses that reward to fine-tune the language model weights. Using this fine-tuning technique, with 24,000 sentences from a science fiction plot summary dataset, halves the percentage of generated text containing non-normative behavior from 35.1% to 15.7%.

Text

Controllable Neural Story Plot Generation via Reward Shaping

WriterForcing: Generating more interesting story endings

Guided Neural Language Generation for Automated Storytelling

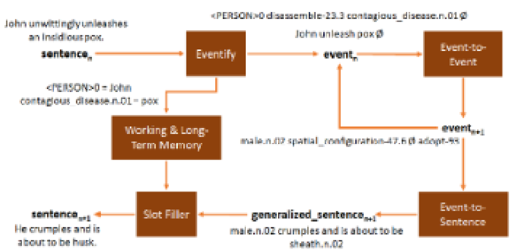


Figure 1: The full automated story generation pipeline illustrating an example where the event-to-event module generates only a single following event.

Automatic Detection of Narrative Structure for High-Level Story Representation

2 Propp's Morphology

In his book, *Morphology of the Folktale*, Vladimir Propp [27] analysed a subset of the corpus of Russian folk tales compiled by Afanasyev. Over a set of 100 tales, Propp identified five categories of elements which he claims define a tale as a whole.

1. Character Functions - a sequence of 31 character-based functions; the narrative units of a tale which are performed by the *dramatis personae*.
2. Conjunctive Elements - when successive functions are performed by different characters, the latter character must somehow be in-

formed of everything that has occurred up until that point. This may for example occur when characters act *ex machina*, are all knowing, or overhear a dialogue between others.

3. Character Motivations - the goals and aims of characters, which drive their actions.
4. Character Appearance - the forms in which characters first enter the story, for example an accidental encounter, or sudden arrival.
5. Attributive Elements - the specific qualities belonging to each

Toward Automated Quest Generation in Text-Adventure Games

Interactive fictions, or text-adventures, are games in which a player interacts with a world entirely through textual descriptions and text actions. Text-adventure games are typically structured as puzzles or quests wherein the player must execute certain actions in a certain order to succeed. In this paper, we consider the problem of procedurally generating a quest, defined as a series of actions required to progress towards a goal, in a text-adventure game. Quest generation in text environments is challenging because they must be semantically coherent. We present and evaluate two quest generation techniques: (1) a Markov model, and (2) a neural generative model. We specifically look at generating quests about cooking and train our models on recipe data. We evaluate our techniques with human participant studies looking at perceived creativity and coherence.

Automated generation of text-adventure games can broadly be split into two considerations: (1) the structure of the world, including the layout of rooms, textual description of rooms, objects, and non-player characters; and (2) the quest, consisting of the partial ordering of activities that the player must engage in to make progress toward the end of the game. In this work, we focus on meth-

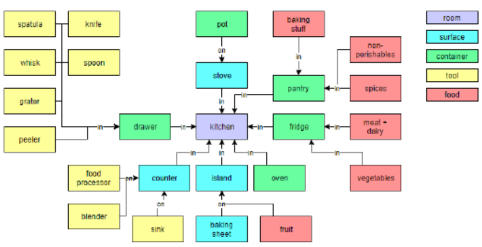


Figure 3: Object graph in the one room map.

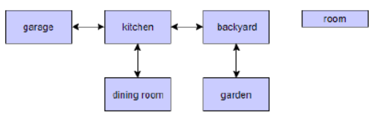
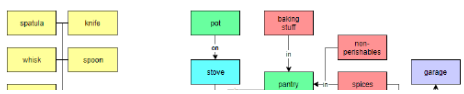


Figure 4: Room layout in the five room map.



gredients and instructions in the game world. This requires us to first determine the structure of the game world and the locations of objects within this world in addition to transforming the set of generated instructions into executable actions. We use two types of semantically grounded knowledge graphs to represent this information: the ob-

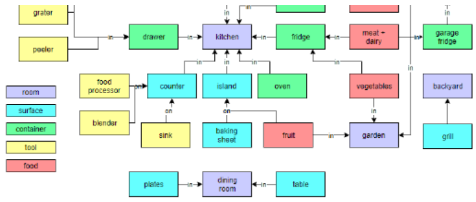


Figure 5: Object graph in the five room map.

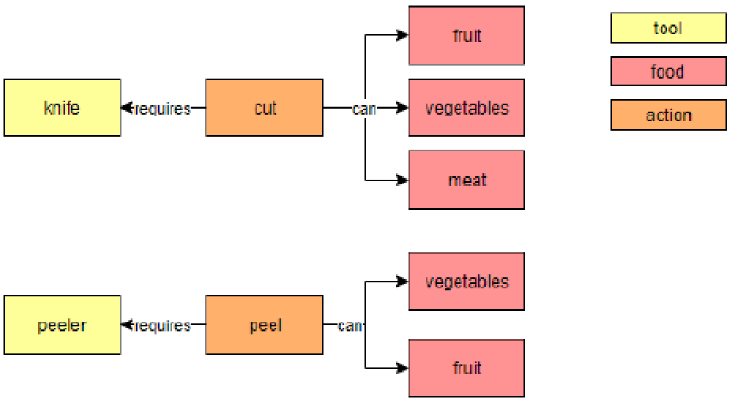


Figure 6: Example action graph for both maps.

The object graph for the 1R map as shown in Fig. 3 is largely inspired by the simple, pre-built game provided within TextWorld (Côté et al., 2018). This object graph determines how and

We present results for four metrics: coherence, unpredictability (or surprise), novelty (or originality), and value (or accomplishment) for each of the games. Additionally, we also show the p-value

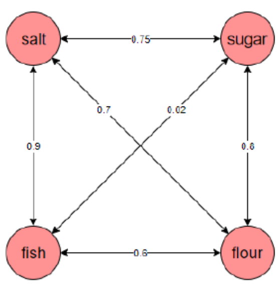


Figure 1: Example of ingredient connections.

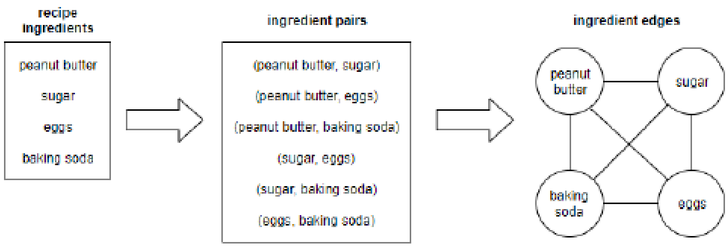


Figure 2: Ingredient extraction process.

Bringing Stories Alive: Generating Interactive Fiction Worlds

3 World Generation

World generation happens in two phases. In the first phase, a partial knowledge graph is extracted from a story plot and then filled in using thematic commonsense knowledge. In the second phase, the graph is used as the skeleton to generate a full interactive fiction game—generating textual descriptions or "flavortext" for rooms and embedded objects. We present a novel neural approach in addition to a rule guided baseline for each of these phases in this section.

World building forms the foundation of any task that requires narrative intelligence. In this work, we focus on procedurally generating interactive fiction worlds—text-based worlds that players "see" and "talk to" using natural language. Generating these worlds requires referencing everyday and thematic commonsense priors in addition to being semantically consistent, interesting, and coherent throughout. Using existing story plots as inspiration, we present a method that first extracts a partial knowledge graph encoding basic information regarding world structure such as locations and objects. This knowledge graph is then automatically completed utilizing thematic knowledge and used to guide a neural language generation model that fleshes out the rest of the world.We perform human participant-based evaluations, testing our neural model's ability to extract and fill-in a knowledge graph and to generate language conditioned on it against rule-based and human-made baselines. Our code is available at https://github.com/rajammanabrolu/WorldGeneration.

In this paper we explore a different challenge for artificial intelligence: automatically generating text-based virtual worlds for interactive fictions. A core component of many narrative-based tasks—everything from storytelling to game generation—is world building. The world of a story or game defines the boundaries of where the narrative is allowed and what the player is allowed to do. There are four core challenges to world generation: (1) commonsense knowledge: the world must reference priors that the player possesses so that players can make sense of the world and build expectations on how to interact with it. This is especially true in interactive fictions where the world is presented textually because many details of the world necessarily be left out (e.g., the pot is on a stove; kitchens are found in houses) that might otherwise be literal in a graphical virtual world. (2) Thematic knowledge: interactive fictions usually involve a theme or genre that comes with its own expectations. For example, light speed travel is plausible in sci-fi worlds but not realistic in the real world. (3) Coherence: the world must not appear to be an random assortment of locations. (3) Natural language: The descriptions of the rooms as well as the permissible actions must text, implying that the system has natural language generation capability.

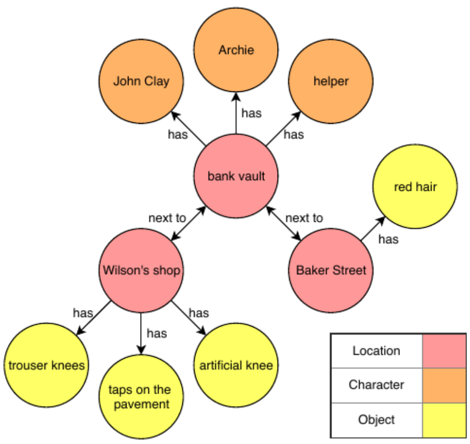


Figure 2: Example knowledge graph constructed by AskBERT.

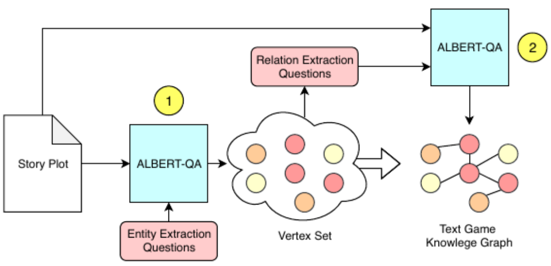


Figure 3: Overall AskBERT pipeline for graph construction.

Bank vault
It is about three feet in height, and one and a half in width.
Exits: Baker Street and Wilson's shop
You see: Archie, Helper and John Clay

Action: **Examine John Clay**

John Clay
Short, stocky, and one of the taller kind, John Clay is the kind of man who lives and dies by the watch he keeps.

Action: **Go to Baker Street**

Baker Street
Like any other street in London, it is a-stage-set, with the best and the worst of society crammed into one place.

Figure 1: Example player interaction in the deep neural generated mystery setting.

Induction and Reference of Entities in a Visual Story

We are enveloped by stories of visual interpretations in our everyday lives. The way we narrate a story often comprises of two stages, which are, forming a central mind map of entities and then weaving a story around them. A contributing factor to coherence is not just basing the story on these entities but also, referring to them using appropriate terms to avoid repetition. In this paper, we address these two stages of introducing the right entities at seemingly reasonable junctures and also referring them coherently in the context of visual storytelling. The building blocks of the central mind map, also known as entity skeleton are entity chains including nominal and coreference expressions. This entity skeleton is also represented in different levels of abstractions to compose a generalized frame to weave the story. We build upon an encoder-decoder framework to penalize the model when the decoded story does not adhere to this entity skeleton. We establish a strong baseline for skeleton informed generation and then extend this to have the capability of *multitasking* by predicting the skeleton in addition to generating the story. Finally, we build upon this model and propose a *glocal hierarchical attention model* that attends to the skeleton both at the sentence (local) and the story (global) levels. We observe that our proposed models outperform the baseline in terms of automatic evaluation metric, METEOR. We perform various analysis targeted to evaluate the performance of our task of enforcing the entity skeleton such as the number and diversity of the entities generated. We also conduct human evaluation from which it is concluded that the visual stories generated by our model are preferred 82% of the times. In addition, we show that our glocal hierarchical attention model improves coherence by introducing more pronouns as required by the presence of nouns.

Character-Centric Storytelling

Sequential vision-to-language or visual storytelling has recently been one of the areas of focus in computer vision and language modeling domains. Though existing models generate narratives that read subjectively well, there could be cases when these models miss out on generating stories that account and address all prospective human and animal characters in the image sequences. Considering this scenario, we propose a model that implicitly learns relationships between provided characters and thereby generates stories with respective characters in scope. We use the VIST dataset for this purpose and report numerous statistics on the dataset. Eventually, we describe the model, explain the experiment and discuss our current status and future work.

A Character-Centric Neural Model for Automated Story Generation

Automated story generation is a challenging task which aims to automatically generate convincing stories composed of successive plots correlated with consistent characters. Most recent generation models are built upon advanced neural networks, e.g., variational autoencoder, generative adversarial network, convolutional sequence to sequence model. Although these models have achieved prompting results on learning linguistic patterns, very few methods consider the attributes and prior knowledge of the story genre, especially from the perspectives of explainability and consistency. To fill this gap, we propose a character-centric neural storytelling model, where a story is created encircling the given character, i.e., each part of a story is conditioned on a given character and corresponded context environment. In this way, we explicitly capture the character information and the relations between plots and characters to improve explainability and consistency. Experimental results on open dataset indicate that our model yields meaningful improvements over several strong baselines on both human and automatic evaluations.

Title (Given)	Rush Hour
Character (Given)	A Police (Represented by a vector)
Context	... A gun battle breaks out.
Predicted Action	Arrest
Generated Sentence	Office Chan arrests a group of gun smugglers.

Table 1: An example of title, character, context, action prediction and sentence generation in our system.
