

Bringing Stories Alive: Generating Interactive Fiction Worlds

Prithviraj Ammanabrolu,* Wesley Cheung,* Dan Tu, William Broniec, Mark O. Riedl

School of Interactive Computing

Georgia Institute of Technology

{raj.ammanabrolu, wcheung8, d.tu, wbroniec3, riedl}@gatech.edu

Abstract

Interactive fictions—also called text-based games—are games in which a player interacts with a virtual world purely through textual natural language. In this work, we focus on procedurally generating interactive fiction worlds. Generating these worlds requires (a) referencing everyday and thematic commonsense priors in addition to (b) being semantically consistent, (c) interesting, (d) coherent throughout, all while (e) producing fluent natural language descriptions of places, people, and things. Using existing story plots from books 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 evaluate generated worlds with human-participant studies, comparing our technique against rule-based and human-made baselines¹.

Introduction

Interactive fictions—also called text-adventure games or text-based games—are games in which a player interacts with a virtual world purely through textual natural language—receiving descriptions of what they “see” and writing out how they want to act. An example can be seen in Figure 1. Interactive fiction games are often structured as puzzles, or quests, set within the confines of given game world. Interactive fictions have been adopted as a test-bed for real-time game playing agents (Narasimhan, Kulkarni, and Barzilay 2015; Côté et al. 2018; Hausknecht et al. 2019b). Unlike other, graphical games, interactive fictions test agents’ abilities to infer the state of the world through communication and to affect change in the world through language. Interactive fictions are typically modeled after real or fantasy worlds; commonsense knowledge is an important factor in successfully playing interactive fictions (Ammanabrolu and Riedl 2019a; Yin and May 2019).

In this paper we explore a different challenge for artificial intelligence: *automatically generating text-based virtual worlds for interactive fictions*. There are four core challenges to text 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, in fantasy worlds, animals can often talk. (3) Coherence: the world must not appear to be a random assortment of locations. (4) Natural language: The descriptions of the rooms as well as the permissible actions must be text, implying that the system has natural language fluency.

We present an approach to text world generation that constructs an interactive, playable, and explorable world based on the plot of a book. Specifically, we take a story such as Sherlock Holmes or Rapunzel—a normally linear reading experience. This provides thematic and genre priors. It also makes it easier to evaluate generated worlds by providing an implied ground truth.

Our method extracts a partial, partially-connected knowledge graph from the input story plot, encoding information regarding locations, characters, and objects, and the relations between them. However, stories often do not explicitly contain all the information required to fully fill out such a graph. Our method fills in missing relation and affordance—the often unstated knowledge on how to use objects—information using thematic knowledge gained from training on stories in a similar genre. The completed knowledge graph is used to guide a text description generation process for the locations, characters, and objects. The game is assembled from the knowledge graph and corresponding generated descriptions.

We performed two separate human subject evaluations—one for the first phase of knowledge graph construction and another for the overall game creation process—testing specifically for coherence, interestingness, and the ability to

*Indicates equal contribution

Copyright © 2020, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

¹Code and data can be found at <https://github.com/rajammanabrolu/WorldGeneration>

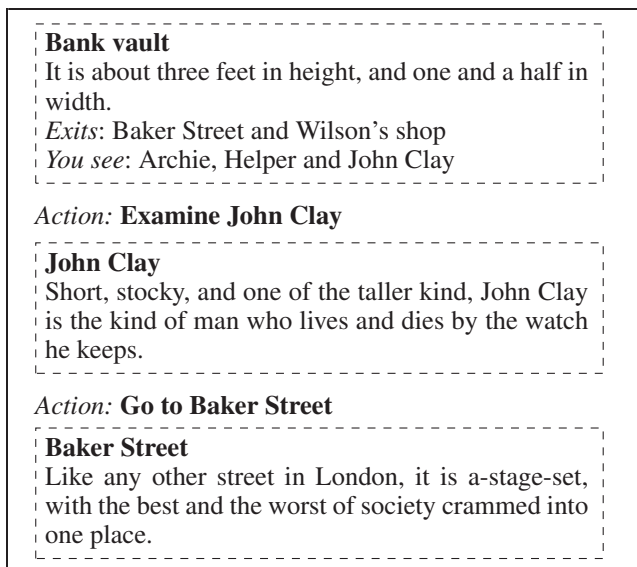


Figure 1: Example player interaction in the deep neural generated mystery setting.

maintain a theme or genre.

Related Work

There have been a number of recent works in developing agents that can play text games (Narasimhan, Kulkarni, and Barzilay 2015; Haroush et al. 2018; Côté et al. 2018; Hausknecht et al. 2019a). Ammanabrolu and Riedl (2019a; 2019b) and Ammanabrolu and Hausknecht (2020) in particular use knowledge graphs as state representations for game-playing agents. These works all focus on agents that learn to play a given set of interactive fiction games as opposed to generating them.

Permar and Magerko (2013) present a method of generating cognitive scripts required for freeform activities in the form of imaginative play. They take existing cognitive scripts and blend them in the vein of more traditional conceptual blending (Veale, O’donoghue, and Keane 2000; Zook, Magerko, and Riedl 2011) to create new blended scripts. This work does not learn to generate content for interactive fiction games but rather make content by recombining existing, pre-defined knowledge.

Scheherazade-IF (Guzdial et al. 2015) is a system that generates choose-your-own-adventure style hyperlinked interactive narratives. It learns a graph of key plot elements and temporal relations between them based on stories written by crowd sourcing the task of writing short stories (Li et al. 2012). Womack and Freeman (2019) propose a method of creating interactive narratives revolving around locations, wherein sentences are mapped to a real-world GPS location from a corpus of sentences belonging to a certain genre. In contrast to these models, our method generates a parser-based interactive fiction in which the player types in a textual command, allowing for greater expressiveness.

Ammanabrolu et al. (2020) present a technique for gener-

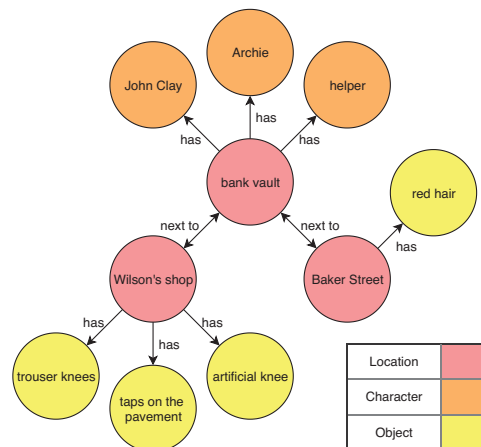


Figure 2: Example knowledge graph constructed by AskBERT.

ating *quests* for text-based games that learns from a corpus of quests and other descriptions of procedural activity and then grounded into a given, hand-authored text world. This paper focuses on the problem of *world* generation glossed in the prior work; these two systems are complimentary.

Light (Urbanek et al. 2019) is a crowdsourced dataset of grounded text-adventure game dialogues. It contains information regarding locations, characters, and objects set in a fantasy world. Fan et al. (2019) use Light to generate worlds for text-based games. They train a neural network based model using Light to compositionally arrange locations, characters, and objects into an interactive world. In contrast, we focus on extracting the information necessary for building interactive worlds from existing story plots.

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.

Knowledge Graph Construction

The first phase is to extract a knowledge graph from the story that depicts locations, characters, objects, and the relations between these entities. We present two techniques. The first uses neural question-answering technique to extract relations from a story text. The second, provided as a baseline, uses OpenIE5,² a commonly used rule-based information extraction technique. For the sake of simplicity, we considered primarily the location-location and location-character/object relations, represented by the “next to” and “has” edges respectively in Figure 2.

²<https://github.com/dair-iitd/OpenIE-standalone>

Neural Graph Construction While many neural models already exist that perform similar tasks such as named entity extraction and part of speech tagging, they often come at the cost of large amounts of specialized labeled data suited for that task. We instead propose a new method that leverages models trained for context-grounded question-answering tasks to do entity extraction with no task dependent data or fine-tuning necessary. Our method, dubbed *AskBERT*, leverages the Question-Answering (QA) model ALBERT (Lan et al. 2019) by *asking* it relevant questions. AskBERT consists of two steps seen in Figure 3: vertex extraction and graph construction.

The first step is to extract the set of entities—graph vertices—from the story. We are looking to extract information specifically regarding characters, locations, and objects. This is done by using asking the QA model questions such as “Who is a character in the story?”. Ribeiro, Guestrin, and Singh (2019) have shown that the phrasing of questions given to a QA model is important and this forms the basis of how we formulate our questions—questions are asked so that they are more likely to return a single answer, e.g. asking “Where is a location in the story?” as opposed to “Where are the locations in the story?”. In particular, we notice that pronoun choice can be crucial; “Where is a location in the story?” yielded more consistent extraction than “What is a location in the story?”. ALBERT QA is trained to also output a special *<no-answer>* token when it cannot find an answer to the question within the story. Our method makes use of this by iteratively asking QA model a question and masking out the most likely answer outputted on the previous step. This process continues until the *<no-answer>* token becomes the most likely answer.

The next step is graph construction. Typical interactive fiction worlds are usually structured as trees, i.e. no cycles except between locations. Using this fact, we use an approach that builds a graph from the vertex set by one relation—or edge—at a time. Once again using the entire story plot as context, we query the ALBERT-QA model picking a random starting location x from the set of vertices previously extracted and asking the questions “What location can I visit from x ?” and “Who/What is in x ?”. The methodology for phrasing these questions follows that described for the vertex extraction. The answer given by the QA model is matched to the vertex set by picking the vertex u that contains the best word-token overlap with the answer. Relations between vertices are added by computing a relation probability on the basis of the output probabilities of the answer given by the QA model. The probability that vertices x, u are related:

$$P(x, u) = \frac{p(x, u) + p(u, x)}{2} \quad (1)$$

where

$$p(x, u) = \sum_{o \in \text{QA outputs}} p(o) \mathbb{1}\{u = \underset{v}{\operatorname{argmax}}(v \cap o)\} \quad (2)$$

is the sum of the individual token probabilities of the overlapping tokens in the answer from the QA model and u .

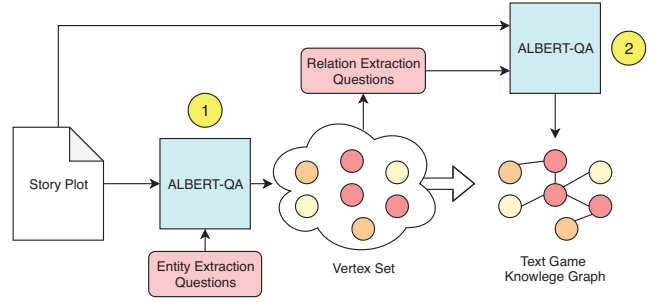


Figure 3: Overall AskBERT pipeline for graph construction.

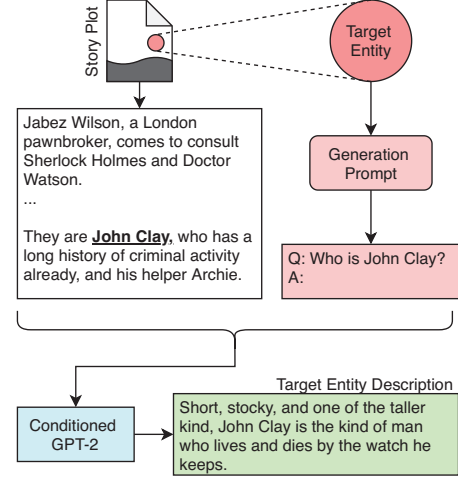


Figure 4: Overview for neural description generation.

Rule-Based Graph Construction We compared our proposed AskBERT method with a non-neural, rule-based approach. This approach is based on the information extracted by OpenIE5, followed by some post-processing such as named-entity recognition and part-of-speech tagging. OpenIE5 combines several cutting-edge ideas from several existing papers (Saha and Mausam 2018; Pal and Mausam 2016; Christensen et al. 2011) to create a powerful information extraction tool. For a given sentence, OpenIE5 generates multiple triples in the format of $\langle \text{entity}, \text{relation}, \text{entity} \rangle$ as concise representations of the sentence, each with a confidence score. These triples are also occasionally annotated with location information indicating that a triple happened in a location.

As in the neural AskBERT model, we attempt to extract information regarding locations, characters, and objects. The entire story plot is passed into the OpenIE5 and we receive a set of triples. The location annotations on the triples are used to create a set of locations. We mark which sentences in the story contain these locations. POS tagging based on marking noun-phrases is then used in conjunction with NER to further filter the set of triples—identifying the set of characters and objects in the story.

The graph is constructed by linking the set of triples on the basis of the location they belong to. While some sentences contain very explicit location information for OpenIE5 to

mark it out in the triples, most of them do not. We therefore make the assumption that the location remains the same for all triples extracted in between sentences where locations are explicitly mentioned. For example, if there exists *locationA* in the 1st sentence and *locationB* in the 5th sentence of the story, all the events described in sentences 1-4 are considered to take place in *locationA*. The entities mentioned in these events are connected to *locationA* in the graph.

Description Generation

The second phase involves using the constructed knowledge graph to generate textual descriptions of the entities we have extracted, also known as flavortext. This involves generating descriptions of what a player “sees” when they enter a location and short blurbs for each object and character. These descriptions need to not only be faithful to the information present in the knowledge graph and the overall story plot but to also contain flavor and be interesting for the player.

Neural Description Generation Here, we approach the problem of description generation by taking inspiration from conditional transformer-based generation methods (Shirish Keskar et al. 2019) and use our constructed knowledge graph to aid in generating the textual descriptions that make up the world. Our approach is outlined in Figure 4 and an example description shown in Figure 1. The knowledge graph is crucial in this process for two main reasons: (1) it lends thematically consistency to the structure of the world—the world would simply be a random collection of locations otherwise, and (2) the relations in the knowledge graph are used to construct the prompt used to condition the description generation network. For any given entity in the story, we first locate it in the story plot and then construct a prompt which consists of the entire story up to and including the sentence when the entity is first mentioned in the story followed by a question asking to describe that entity. With respect to prompts, we found that more direct methods such as question-answering were more consistent than open-ended sentence completion. For example, “Q: Who is the prince? A:” often produced descriptions that were more faithful to the information already present about the prince in the story than “You see the prince. He is/looks”. This question depends on the relations linking the entity to others in the knowledge graph as seen in Figure 2, e.g. “next to” refer to locations and so lead to prompt such as “What does the bank vault look like?”.

For our transformer-based generation, we use a pre-trained 355M GPT-2 model (Radford et al. 2019) finetuned on a corpus of plot summaries collected from Wikipedia. The plots used for finetuning are tailored specific to the genre of the story in order to provide more relevant generation for the target genre. Additional details regarding the datasets used are provided in the section on Evaluation. This method strikes a balance between knowledge graph verbalization techniques which often lack “flavor” and open ended generation which struggles to maintain semantic coherence.

Rules-Based Description Generation In the rule-based approach, we utilized the templates from the built-in text game generator of TextWorld (Côté et al. 2018) to generate

the description for our graphs. TextWorld is an open-source library that provides a way to generate text-game learning environments for training reinforcement learning agents using pre-built grammars.

Two major templates involved here are the Room Intro Templates and Container Description Templates from TextWorld, responsible for generating descriptions of locations and blurbs for objects/characters respectively. The location and object/character information are taken from the knowledge graph constructed previously. Examples of:

- Room Intro Templates: “This might come as a shock to you, but you’ve just *#entered#* a *<location-name>*”
- Container Description Templates: “The *<location-name>* *#contains#* *<object/person-name>*”

Each token surrounded by # sign can be expanded using a select set of terminal tokens. For instance, *#entered#* could be filled with any of the following phrases here: entered; walked into; fallen into; moved into; stumbled into; come into. Additional prefixes, suffixes and adjectives were added to increase the relative variety of descriptions. Unlike the neural methods, the rule-based approach is not able to generate detailed and flavorful descriptions of the properties of the locations/objects/characters. By virtue of the templates, however, it is much better at maintaining consistency with the information contained in the knowledge graph.

Evaluation

We conducted two sets of human participant evaluations by recruiting participants over Amazon Mechanical Turk. The first evaluation tests the knowledge graph construction phase, in which we measure perceived coherence and genre or theme resemblance of graphs extracted by different models. The second study compares full games—including description generation and game assembly, which can’t easily be isolated from graph construction—generated by different methods. This study looks at how interesting the games were to the players in addition to overall coherence and genre resemblance. Both studies are performed across two genres: mystery and fairy-tales. This is done in part to test the relative effectiveness of our approach across different genres with varying thematic commonsense knowledge. The dataset used was compiled via story summaries that were scraped from Wikipedia via a recursive crawling bot. The bot searched pages for both plot sections as well as links to other potential stories. From the process, 695 fairy-tales and 536 mystery stories were compiled from two categories: novels and short stories. We note that the mysteries did not often contain many fantasy elements, i.e. they consisted of mysteries set in our world such as Sherlock Holmes, while the fairy-tales were much more removed from reality. Details regarding how each of the studies were conducted and the corresponding setup are presented below.

Knowledge Graph Construction Evaluation

We first select a subset of 10 stories randomly from each genre and then extract a knowledge graph using three different models. Each participant is presented with the three

Genre	Category	Neural	Rules
Mystery	Locations	7.2	3.5
	Characters	4.8	4.1
	Objects	3.2	12.2
Fairy-tale	Locations	4.0	1.8
	Characters	3.3	1.2
	Objects	4.1	8.7

Table 1: Vertex statistics: Average vertex count by type per genre. The random model has the same vertex statistics as the neural model.

Genre	Statistic	Neural	Rules	Random
Mystery	Avg. Edges	10.7	22.3	10.7
	Avg. Degree	1.63 ± 1.77	2.15 ± 0.38	1.63 ± 1.63
Fairy-tale	Avg. Edges	16.7	12	16.7
	Avg. Degree	1.73 ± 2.04	1.98 ± 0.29	1.73 ± 1.64

Table 2: Edge and degree statistics: Average edge count , average degree count, and degree standard deviation of the graphs.

graphs extracted from a single story in each genre and then asked to rank them on the basis of how coherent they were and how well the graphs match the genre. The graphs resembles the one shown in Figure 2 and are presented to the participant sequentially. The exact order of the graphs and genres was also randomized to mitigate any potential latent correlations. Overall, this study had a total of 130 participants. This ensures that, on average, graphs from every story were seen by 13 participants.

In addition to the neural AskBERT and rules-based methods, we also test a variation of the neural model which we dub to be the “random” approach. The method of vertex extraction remains identical to the neural method, but we instead connect the vertices randomly instead of selecting the most confident according to the QA model. We initialize the graph with a starting location entity. Then, we randomly sample from the vertex set and connect it to a randomly sampled location in the graph until every vertex has been connected. This ablation in particular is designed to test the ability of our neural model to predict relations between entities. It lets us observe how accurately linking related vertices effects each of the metrics that we test for. For a fair comparison between the graphs produced by different approaches, we randomly removed some of the nodes and edges from the initial graphs so that the maximum number of locations per graph and the maximum number of objects/people per location in each story genre are the same.

The results are shown in Table 3. We show the median rank of each of the models for both questions across the genres. Ranked data is generally closely interrelated and so we perform Friedman’s test between the three models to validate that the results are statistically significant. This is presented as the p -value in table (asterisks indicate significance at $p < 0.05$). In cases where we make comparisons between specific pairs of models, when necessary, we additionally perform the Mann-Whitney U test to ensure that the rankings differed significantly.

Genre	Questions	Neural	Rules	Random	p-value
Mystery	Resembles Genre	2	1	3	0.35
	Coherence	1	2	3	0.049*
Fairy-tale	Resembles Genre	1	3	2	0.014*
	Coherence	1	3	2	0.013*

Table 3: Results of the knowledge graph evaluation study showing the median rank of each model.

In the mystery genre, the rules-based method was often ranked first in terms of genre resemblance, followed by the neural and random models. This particular result was not statistically significant however, likely indicating that all the models performed approximately equally in this category. The neural approach was deemed to be the most coherent followed by the rules and random. For the fairy-tales, the neural model ranked higher on both of the questions asked of the participants. In this genre, the random neural model also performed better than the rules-based approach.

Tables 1 and 2 show the statistics of the constructed knowledge graphs in terms of vertices and edges. We see that the rules-based graph construction has a lower number of locations, characters, and relations between entities but far more objects in general. The greater number of objects is likely due to the rules-based approach being unable to correctly identify locations and characters. The gap between the methods is less pronounced in the mystery genre as opposed to the fairy-tales, in fact the rules-based graphs have more relations than the neural ones. The random and neural models have the same number of entities in all categories by construction but random in general has lower variance on the number of relations found. In this case as well, the variance is lower for mystery as opposed to fairy-tales. When taken in the context of the results in Table 3, it appears to indicate that leveraging thematic commonsense in the form of AskBERT for graph construction directly results in graphs that are more coherent and maintain genre more easily. This is especially true in the case of the fairy-tales where the thematic and everyday commonsense diverge more than in the case of the mysteries.

Full Game Evaluation

This participant study was designed to test the overall game formulation process encompassing both phases described in the section on World Generation. A single story from each genre was chosen by hand from the 10 stories used for the graph evaluation process. From the knowledge graphs for this story, we generate descriptions using the neural, rules, and random approaches described previously. Additionally, we introduce a human-authored game for each story here to provide an additional benchmark. This author selected was familiar with text-adventure games in general as well as the genres of detective mystery and fairy tale. To ensure a fair comparison, we ensure that the maximum number of locations and maximum number of characters/objects per location matched the other methods. After setting general format expectations, the author read the selected stories and constructed knowledge graphs in a corresponding three step process of: identifying the n most important entities in the

Genre	Questions	Neural vs. Random (%)	Neural vs. Rules (%)	Neural vs. Human (%)
Mystery	Interesting	55 vs. 45	28 vs. 72*	31 vs. 69*
	Coherence	64 vs. 36*	55 vs. 45*	31 vs. 69*
	Resembles Genre	55 vs. 45	62 vs. 38*	25 vs. 75*
Fairy-tale	Interesting	58 vs. 42	63 vs. 37*	36 vs. 64*
	Coherence	75 vs. 25*	75 vs. 25*	55 vs. 45
	Resembles Genre	75 vs. 25*	63 vs. 37*	31 vs. 69*

Table 4: Results of the full game evaluation participant study showing the pairwise percentage of participants preferring the baseline shown to the neural model. *Indicates statistical significance ($p < 0.05$).

story, mapping positional relationships between entities, and then synthesizing flavor text for the entities based off of said location, the overall story plot, and background topic knowledge.

Once the knowledge graph and associated descriptions are generated for a particular story, they are then automatically turned into a fully playable text-game using the text game engine Evennia.³ Evennia was chosen for its flexibility and customization, as well as a convenient web client for end user testing. The data structures were translated into builder commands within Evennia that constructed the various layouts, flavor text, and rules of the game world. Users were placed in one “room” out of the different world locations within the game they were playing, and asked to explore the game world that was available to them. Users achieved this by moving between rooms and investigating objects. Each time a new room was entered or object investigated, the player’s total number of explored entities would be displayed as their score.

Each participant was asked to play the neural game and then another one from one of the three additional models within a genre. The completion criteria for each game is collect half the total score possible in the game, i.e. explore half of all possible rooms and examine half of all possible entities. This provided the participant with multiple possible methods of finishing a particular game. On completion, the participant was asked to rank the two games according to overall perceived coherence, interestingness, and adherence to the genre. We additionally provided a required initial tutorial game which demonstrated all of these mechanics. The order in which participants played the games was also randomized as in the graph evaluation to remove potential correlations. We had 75 participants in total, 39 for mystery and 36 for fairy-tales. As each player played the neural model created game and one from each of the other approaches—this gave us 13 on average for the other approaches in the mystery genre and 12 for fairy-tales.

The summary of the results of the full game study is shown in Table 4. As the comparisons made in this study are all made pairwise between our neural model and one of the baselines—they are presented in terms of what percentage of participants prefer the baseline game over the neural game. Once again, as this is highly interrelated ranked data, we perform the Mann-Whitney U test between each of the pairs to ensure that the rankings differed significantly. This

is also indicated on the table.

In the mystery genre, the neural approach is generally preferred by a greater percentage of participants than the rules or random. The human-made game outperforms them all. A significant exception is that participants thought that the rules-based game was more interesting than the neural game. The trends in the fairy-tale genre are in general similar with a few notable deviations. The first deviation is that the rules-based and random approaches perform significantly worse than neural in this genre. We see also that the neural game is as coherent as the human-made game.

As in the previous study, we hypothesize that this is likely due to the rules-based approach being more suited to the mystery genre, which is often more mundane and contains less fantastical elements. By extension, we can say that thematic commonsense in fairy-tales has less overlap with everyday commonsense than for mundane mysteries. This has a few implications, one of which is that this theme specific information is unlikely to have been seen by OpenIE5 before. This is indicated in the relatively improved performance of the rules-based model in this genre across in terms of both interestingness and coherence. The genre difference can also be observed in terms of the performance of the random model. This model is also lacking when compared to our neural model across all the questions asked especially in the fairy-tale setting. This implies that filling in gaps in the knowledge graph using thematically relevant information from AskBERT results in more interesting and coherent descriptions and games especially in settings where the thematic commonsense diverges from everyday commonsense.

Conclusion

Procedural world generation systems are required to be semantically consistent, comply with thematic and everyday commonsense understanding, and maintain overall interestingness. We describe an approach that transform a linear reading experience in the form of a story plot into an interactive narrative experience. Our method, AskBERT, extracts and fills in a knowledge graph using thematic commonsense and then uses it as a skeleton to flesh out the rest of the world. A key insight from our human participant study reveals that the ability to construct a thematically consistent knowledge graph is critical to overall perceptions of coherence and interestingness particularly when the theme diverges from everyday commonsense understanding.

³<http://www.evennia.com/>

References

- Ammanabrolu, P., and Hausknecht, M. 2020. Graph constrained reinforcement learning for natural language action spaces. In *International Conference on Learning Representations*.
- Ammanabrolu, P., and Riedl, M. 2019a. Transfer in deep reinforcement learning using knowledge graphs. In *TextGraphs-13 at EMNLP-19*, 1–10. Hong Kong: Association for Computational Linguistics.
- Ammanabrolu, P., and Riedl, M. O. 2019b. Playing text-adventure games with graph-based deep reinforcement learning. In *Proceedings of 2019 Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019*.
- Ammanabrolu, P.; Broniec, W.; Mueller, A.; Paul, J.; and Riedl, M. O. 2020. Toward automated quest generation in text-adventure games. In *Proceedings of the International Conference on Computational Creativity ICCI-20*.
- Christensen, J.; Soderland, S.; Etzioni, O.; and Mausam. 2011. An analysis of open information extraction based on semantic role labeling. In *Proceedings of the sixth international conference on Knowledge capture*, 113–120. ACM.
- Côté, M.-A.; Kádár, Á.; Yuan, X.; Kybartas, B.; Barnes, T.; Fine, E.; Moore, J.; Hausknecht, M.; Asri, L. E.; Adada, M.; et al. 2018. Textworld: A learning environment for text-based games. *arXiv preprint arXiv:1806.11532*.
- Fan, A.; Urbanek, J.; Ringshia, P.; Dinan, E.; Qian, E.; Karamcheti, S.; Prabhumoye, S.; Kiela, D.; Rocktaschel, T.; Szlam, A.; et al. 2019. Generating interactive worlds with text. *arXiv preprint arXiv:1911.09194*.
- Guzdial, M.; Harrison, B.; Li, B.; and Riedl, M. 2015. Crowdsourcing open interactive narrative. In *FDG*.
- Haroush, M.; Zahavy, T.; Mankowitz, D. J.; and Mannor, S. 2018. Learning How Not to Act in Text-Based Games. In *Workshop Track at ICLR 2018*, 1–4.
- Hausknecht, M.; Ammanabrolu, P.; Côté, M.-A.; and Yuan, X. 2019a. Interactive fiction games: A colossal adventure. *CoRR abs/1909.05398*.
- Hausknecht, M.; Loynd, R.; Yang, G.; Swaminathan, A.; and Williams, J. D. 2019b. Nail: A general interactive fiction agent. *CoRR abs/1902.04259*.
- Lan, Z.; Chen, M.; Goodman, S.; Gimpel, K.; Sharma, P.; and Soricut, R. 2019. ALBERT: A Lite BERT for Self-supervised Learning of Language Representations. *arXiv e-prints arXiv:1909.11942*.
- Li, B.; Lee-Urban, S.; Appling, D. S.; and Riedl, M. O. 2012. Crowdsourcing Narrative Intelligence. In *Advances in Cognitive Systems*, volume 1, 1–18.
- Narasimhan, K.; Kulkarni, T.; and Barzilay, R. 2015. Language Understanding for Text-based Games Using Deep Reinforcement Learning. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*.
- Pal, H., and Mausam. 2016. Demonyms and compound relational nouns in nominal open ie. In *Proceedings of the 5th Workshop on Automated Knowledge Base Construction*, 35–39.
- Permar, J., and Magerko, B. 2013. A Conceptual Blending Approach to the Generation of Cognitive Scripts for Interactive Narrative. In *9th AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment (AIIDE’13)*.
- Radford, A.; Wu, J.; Child, R.; Luan, D.; Amodei, D.; and Sutskever, I. 2019. Language models are unsupervised multitask learners.
- Ribeiro, M. T.; Guestrin, C.; and Singh, S. 2019. Are red roses red? evaluating consistency of question-answering models. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 6174–6184.
- Saha, S., and Mausam. 2018. Open information extraction from conjunctive sentences. In *Proceedings of the 27th International Conference on Computational Linguistics*, 2288–2299.
- Shirish Keskar, N.; McCann, B.; Varshney, L. R.; Xiong, C.; and Socher, R. 2019. CTRL: A Conditional Transformer Language Model for Controllable Generation. *arXiv e-prints arXiv:1909.05858*.
- Urbanek, J.; Fan, A.; Karamcheti, S.; Jain, S.; Humeau, S.; Dinan, E.; Rocktäschel, T.; Kiela, D.; Szlam, A.; and Weston, J. 2019. Learning to speak and act in a fantasy text adventure game. In *Proceedings of EMNLP-IJCNLP*, 673–683.
- Veale, T.; O’donoghue, D.; and Keane, M. T. 2000. Computation and Blending. *Cognitive Linguistics* 11(3/4):253–282.
- Womack, J., and Freeman, W. 2019. Interactive narrative generation using location and genre specific context. In *Interactive Storytelling*, 343–347.
- Yin, X., and May, J. 2019. Comprehensible context-driven text game playing. *CoRR abs/1905.02265*.
- Zook, A.; Magerko, B.; and Riedl, M. 2011. Formally modeling pretend object play. In *Proceedings of the 8th ACM Conference on Creativity and Cognition*, 147–156. New York, NY, USA: ACM.