Coherent Story Generation with Structured Knowledge

Congda Ma[1] **Kotaro Funakoshi**[1] **Kiyoaki Shirai**[2] **Manabu Okumura**[1]
[1]Tokyo Institute of Technology
[2]Japan Advanced Institute of Science and Technology
{ma, funakoshi, oku}@lr.pi.titech.ac.jp
kshirai@jaist.ac.jp

Abstract

The emergence of pre-trained language models has taken story generation, which is the task of automatically generating a comprehensible story from limited information, to a new stage. Although generated stories from the language models are fluent and grammatically correct, the lack of coherence affects their quality. We propose a knowledge-based multi-stage model that incorporates the schema, a kind of structured knowledge, to guide coherent story generation. Our framework includes a schema acquisition module, a plot generation module, and a surface realization module. In the schema acquisition module, high-relevant structured knowledge pieces are selected as a schema. In the plot generation module, a coherent plot plan is navigated by the schema. In the surface realization module, conditioned by the generated plot, a story is generated. Evaluations show that our methods can generate more comprehensible stories than strong baselines, especially with higher global coherence and less repetition.

1 Introduction

Story generation is a challenging text generation task that uses a limited text, such as a first sentence, as a clue to form a story. The story must not only be grammatically correct but also have logic and consistency, which is known as coherence. Illogical sentences, unrelated entities, and generic phrases are problems that affect the coherence of generated stories (Holtzman et al., 2020).

Incorporating knowledge into models has shown promising results in the story generation task; external information is provided to the models to capture more features not given in the context. For example, Guan et al. (2020) applied commonsense knowledge to their language model by using pre-training. As shown in Table 1, while their method alleviates the information deficit and produces more fluent sentences, there is still room for improving coherence. In particular, what is needed is a more sophisticated mechanism that teaches how to utilize external knowledge more effectively in the model to control the coherence of generated stories.

Context	tyron is headed home from basketball practice.
Human written	he hears a noise coming from outside of his car. he decides to pull over and see what 's going on. tyron sees a metal spike in his tire. he calls his dad for help and waits.
GPT-2	her son has to take his test. her son has to take the test. **his son, his brother, and their mother were both in school**. her son, his brother, and their mother were both disappointed.
Knowledge pre-trained GPT-2	*he is ready to play. he is also dressed in his big black dress.* when he leaves for the game he is surprised. he is headed home.

Table 1: Stories generated by human and neural models (Guan et al., 2020). Generated stories are *illogical*, describe **unrelated entities**, and contain unnatural repetitions. Low global coherence makes automatically generated stories difficult to understand in spite of their sentences being grammatically correct.

To obtain better coherence, many previous studies have attempted to decompose the story generation task into stages. The use of a plot has been shown to help the model understand narratives by providing expectations, resolving ambiguity, and filling in unstated information (Sakaguchi et al., 2021). A script is introduced, which represents a core plot for a story, to guide the surface realization of the story(Fan et al., 2018; Yao et al., 2019). They first predicted the script and then utilized it to generate sentences in a story. In this two-stage generation process, these models generated sentences capturing the lexical information from the plot. However, they did not explore how to have a structure within the plot. The lack of a structure may cause illogical or repeated events to be generated for a plot. As a result, even though each generated sentence was related to the corre-

Proceedings of Recent Advances in Natural Language Processing, pages 681–690
Varna, Sep 4–6, 2023
https://doi.org/10.26615/978-954-452-092-2_074

Figure 1: Effect of a schema on plot generation. Structured knowledge is used to guide plot generation. Compared with the model without a schema, our model generates a more logical plot that is not repetitive.

sponding plot, the coherence between sentences was poor (Fan et al., 2018; Yao et al., 2019). To alleviate this issue, a kind of structured knowledge is desired to be incorporated to drive the plot.

In this paper, we propose a structured knowledge-based multi-stage story generation model. For enhancing the coherence of generated stories, we apply relevant external knowledge as a schema to the plot generation stage to explicitly guide the generation of a plot. The coherent plot can be an excellent navigator that guides the model to generate stories containing more coherent and explainable content.

The aforementioned schema is a concept in psychology that describes a pattern of thought or behavior that organizes categories of information and their relationships to guide perception, interpretation, imagination, or problem solving (APA Dictionary, 2022). "Background knowledge" or "prior knowledge" are also be used interchangeably with schema (Sadoski et al., 1991). They serve a crucial role in providing an account of how old knowledge interacts with new knowledge in perception, language, thought, and memory (Brewer and Nakamura, 1984). There is a clear link between schema and comprehension because a structure facilitates the planful retrieval of textual information and allows the reconstruction of elements that have not been learned or forgotten (Anderson and Pearson, 1984). We consider that schema could provide a window into how models might use knowledge effectively. Encouraged by the concept, We try to apply schema into the model to guide the coherent story generation. Our model utilizes highly relevant knowledge as structured knowledge to compose a schema. The knowledge in the schema could provide external information and stimulate knowledge stored in the model. As shown in Figure 1, when our model infers a plot, it is affected by the schema (*get fire starter*, *gather wood*, and *make fire*). Compared with the event (*build a fire*) predicted by a model without schema, our model can generate a more explainable prediction (*buy some wood and a fire starter*), that is not repetitive. Obviously, a story produced from a coherent plot will be more coherent.

The main contributions of this paper are summarized as follows:

- We construct a multi-stage story generation model by combining BART (Lewis et al., 2020) with GPT-2 (Radford et al., 2019) to generate a coherent story.
- We propose a novel plot generation framework by allowing the incorporation of structured external knowledge into the model. In this model, the schema is utilized to guide the prediction of a coherent plot, thereby improving the coherence of generated stories.
- We develop two models, one with a story-level schema and the other with a sentence-level schema, to explore their ability and limitation of using knowledge in the story generation model.
- The results of objective and subjective evaluations show that our story-level model can generate more coherent stories than strong baselines.

2 Related Work

2.1 Storytelling

Storytelling consists of tasks that aim to generate a readable story like human-writing. Chandu et al. (2019) transformed stories to fit different character styles. Some work tries to generate stories from various sources, including generating a story from a short sequence (Fan et al., 2018; Rashkin et al., 2020), and a topic (Zhai et al., 2019; Yao et al., 2019). While storytelling has developed rapidly in recent years, the quality gap between automatically generated stories and human-written stories is still large.

2.2 Script-based Generation

Script-based story generation is a strategy that decomposes the story generation task into stages. One of the common methods is applying a two-stage model which generates a script that represents a core plot for a story first, then uses the script to guide the surface realization of the story(Fan et al., 2018, 2019; Yao et al., 2019). In Yao et al. (2019) they utilized a storyline before a whole story is generated, which increases the coherence. Xu et al. (2018) proposed a method that uses a compressed sentence as a representation to enrich and control the content of sentences in a generated story. To generate scripts with correct orders, a new dataset "proScript" is created for the scripts generation task (Sakaguchi et al., 2021). Ammanabrolu et al. (2020) proposed an ensemble-based system that can generate semantically-related sentences from scripts (Sakaguchi et al., 2021).

2.3 Knowledge-based Text Generation

Incorporating knowledge has demonstrated advantages in various NLP generation tasks, such as fact-aware generation (Logan et al., 2019), conversation generation (Wang et al., 2020). Especially in open-domain generation tasks, which suffer from the lack of external information, the knowledge provides information that cannot be found in the source and helps the model capture more details. With the development of pre-trained language models, researchers have come to incorporate external knowledge into the pre-trained models. Yang et al. (2019a) utilized knowledge to enhance the representations in BERT to improve comprehension. Xiong et al. (2020) proposed a method to encourage pre-trained language models to learn entity-level knowledge when answering questions. Guan et al. (2020) pre-trained GPT-2 with commonsense knowledge to ensure that the model learns the information and generates more fluent and logical stories.

3 Proposed Methods

3.1 Task Setting

Our task is a story completion task, which is to generate the rest of a story $Y = [s_1, s_2, ..., s_i]$ from the first sentence of the story $X = s_0$, where s_i is the i-th generated sentence.

3.2 Model Architecture

In common with the other multi-stage story generation models, we first generate a plot P from e_0. It is a sequence of events $[e_1, e_2, e_3, ..., e_i]$ where each event corresponds to the core information of a sentence. e_0 is pre-extracted from s_0. Then, story Y is completed according to plot P. We use a phrase containing a predicate to represent an event in a sentence because giving an informative representation helps models capture dependencies in the context (Lin et al., 2021). We apply dependency parsing to recognize the root and its object and retain all the words between them. Then, we normalize the root verb to the base form.

Our model involves schema acquisition (SA), plot generation (PG), and surface realization (SR) modules. The SA module is utilized to obtain the structured knowledge as a schema T from a large set of candidate knowledge pieces K. The PG module is formulated as a knowledge-based generation model, where the schema T and the event e_0 are set as input to generate the following events as the plot P. The SR module is a conditional generation model, where the plot P is expanded to the story Y.

We propose two PG models, i.e., a story-level model and a sentence-level model, to explore the ability and limitations of knowledge use in our models. In the story-level model, the whole plot is generated with the same schema. In contrast, the sentence-level model generates a plot event by event with updated schemata.

3.2.1 Schema Acquisition (SA)

In the SA module, the structured knowledge, the schema T, is acquired from a candidate knowledge set K.

In the previous knowledge-incorporated models (Guan et al., 2020; Ji et al., 2020; Liu et al.,

2021), it is unclear how to use a large number of external knowledge pieces because the models do not know which information is more appropriate to be captured for the current story generation step. They do not acquire new knowledge or update the old knowledge as the stories' backgrounds change.

The schema in this study provides entities and their interactions (predicates) that are relevant in the current background and allows the model to capture necessary information, rather than irrelevant information in the previous models, in which the knowledge is fixed without considering the current background (e.g., in a normal concept net, *pan* is related to *cooker*, but in the context of shopping, *pan* may be more relevant to *cashier*). The knowledge pieces in the schema, such as *pay for the pan*, can specifically give relevant information in such a shopping scenario.

First, we obtain the candidate knowledge set K for the event e_0. A candidate knowledge set is a set of knowledge pieces that are relevant to an event. Each knowledge piece is of a phrasal form beginning with a verb (e.g. *get fire starter*, *gather wood*, *make fire*). Since the knowledge piece contains both a predicate and its arguments, it is shown to be useful to improve language understanding and global coherence (Yang et al., 2019b).

We use COMET-ATOMIC2020 (Hwang et al., 2021) to obtain the knowledge. It is a neural knowledge model that can generate relevant knowledge for an input text under specific relationships. We feed the event e_0 as the input and collect the knowledge pieces generated from the model as the candidate knowledge set K. We utilize the relations under the event-centered category, "IsAfter", "HasSubEvent", "IsBefore", "HinderedBy", "Causes", and "xReason", to get the knowledge pieces.

We need to pick out the knowledge pieces with higher relevance and lower noise from the candidate knowledge set to compose the schema. We introduce semantic similarity to realize the function. For encoding, we utilize Sentence-BERT (Reimers and Gurevych, 2019) because it shows better performance than the traditional BERT on the sentence similarity benchmarks.

In practice, we find some candidate knowledge pieces have only slight difference (e.g., *go to a beach* and *go to the beach*). To delete such duplicate knowledge pieces, following Peng et al. (2021), we first calculate the cosine similarity between each pair of two candidate knowledge pieces. We set 80% semantic similarity as our threshold, which means if the score for a pair is higher than 0.8, only one candidate knowledge piece will be left.

Then, the semantic similarity between the event e_0 and each candidate knowledge piece is calculated. We select the top-n candidate knowledge pieces to compose a schema $T = \{t_1, t_2, t_3, ..., t_n\}$, where t_n represents the knowledge piece with the n-th score.

3.2.2 Plot Generation (PG)

In the PG module, the plot P, which represents the backbone of the story, is generated from the schema T and the event e_0.

We fine-tune a BART to generate a sequence of events $[e_1, e_2, e_3, ..., e_i]$ for the plot P, as BART shows better performance in tasks with external knowledge (Liu et al., 2021; Ji et al., 2020).

When training, e_i is pre-extracted from the sentence s_i in a story. The events except e_0 are combined in order as a target plot.

3.2.3 Surface Realization (SR)

In the SR module, by using the first sentence s_0 and the plot P as the prefix, the rest of a story Y is generated.

We fine-tune a GPT-2 (Radford et al., 2019) to implement the SR module because GPT-2 shows excellent ability in conditional generation tasks (Zhipeng et al., 2019).

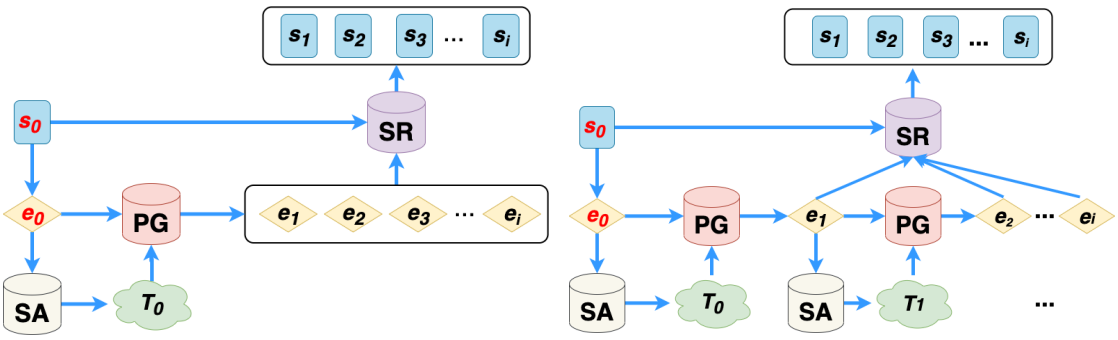
3.3 Plot Generation Strategies

3.3.1 Story-level Model

As shown in Figure 2(a), in the story-level model, we first extract the event e_0 from the first sentence s_0 and then obtain a schema T_0 by SA module. The schema T_0 is utilized for generating the whole plot P.

In the PG module, to help the model recognize different ingredients in the input, we add a special token [k] before every knowledge piece in the schema, and add another special token [e] before the event e_0. These kinds of prompt tokens have been used in related tasks (Gupta and Durrett, 2019; Zheng and Huang, 2021). In the output, we use a special token [sep] between events to distinguish the boundary. [bos] and [eos] tokens are also used to indicate the beginning and end of the output.

When fine-tuning, the form of the source text



(a) (b)

Figure 2: (a): Framework of the story-level model. The whole plot is generated in one iteration. (b): Framework of the sentence-level model. Events in the plot are generated one by one.

and target text is as follows:

$$\text{source}: \text{[k]}\ t_1\ \text{[k]}\ t_2\ \ldots\ \text{[k]}\ t_n\ \text{[e]}\ e_0$$
$$\text{target}: \text{[bos]}\ e_1\ \text{[sep]}\ e_2\ \text{[sep]}\ \ldots\ e_i\ \text{[eos]}$$

where t_n represents the n-th knowledge piece in the schema. The generated plot P concatenated with the first sentence s_0 is fed into the SR module.

In the SR module, we add a special token [e] before each event in the plot and use [sep] to separate the plot P and the first sentence s_0. [bos] and [eos] tokens show the end of the prefixed text and the target text, respectively.

In this strategy, the form of the fine-tuning data for the SR module is:

$$\text{[e]}\ e_1 \ldots \text{[e]} e_i\ \text{[sep]}\ s_0\ \text{[bos]}\ s_1 \ldots s_i\ \text{[eos]};$$

3.3.2 Sentence-level Model

Different from the story-level model, in the sentence-level model, we generate the plot by using a different schema T_{i-1} for each event $e_i (i > 1)$. As shown in Figure 2(b), when generating the event e_i, we rerank the knowledge pieces with the similarity scores to get the updated schema $T_{i-1} = \{t_1^{i-1}, t_2^{i-1}, t_3^{i-1} \ldots t_n^{i-1}\}$, where t_n^{i-1} represents the knowledge piece with the n-th highest score with the event e_{i-1}. Then, we will update it again by e_i in the next step. This procedure is repeated to obtain all the events to combine into a plot. Please note that the initial input is the event e_0 and the schema T_0, as in the story-level model.

In the PG module, as in the story-level model, for the input, we add a special token [k] before every knowledge piece in the schema, and add a special token [e] before the event e_i. In the output, because there is only one event in the output, we only use [bos] and [eos] tokens are added to show the beginning and end of the output.

When fine-tuning, the source text and the target text are:

$$\text{source}: \text{[k]}\ t_1^{i-1}\ \text{[k]}\ t_2^{i-1}\ \ldots\ \text{[k]}\ t_n^{i-1}\ \text{[e]}\ e_{i-1}$$
$$\text{target}: \text{[bos]} e_i \text{[eos]}$$

The SR module in the sentence-level model has the same structure as in the story-level model. The generated plot P concatenated with the first sentence s_0 is used as the input to generate the story.

4 Experiments

In this section, the details of the dataset, the experimental settings, and the baselines in our experiments are introduced.

4.1 Dataset

We used the ROCstory (Mostafazadeh et al., 2016, 2017) and WritingPrompts (Fan et al., 2018) datasets in our experiments. ROCstory dataset contains 98,161 English stories, where each story consists of five sentences. Excluding the stories from which we could not extract events[1], we separated the dataset into 86,892, 4,827, and 4,828 stories for training, validation, and test sets, respectively. In addition, the first letter was replaced with a lowercase letter. For the WritingPrompts dataset, we first randomly sampled 100,000, 5000, and 5000 stories as training, validation, and test datasets, respectively. Then, we used the spaCy library[2] to segment every story into sentences and retained only the first five sentences as a story.

[1] Stories that contain sentences which can not extract predicates.

[2] https://spacy.io/

4.2 Experimental Settings

In our experiments, we use the first sentence as input, and the number of generated sentences was limited to four, following the dataset ($i = 4$). We applied the spaCy library for dependency parsing. We used the parameters of the large version of BART and the small version of GPT-2.[3] The number of knowledge pieces in a schema was tuned to 60 on the validation dataset.

4.3 Baselines

We compared our models with the following story generation models:

Plan & Write (Yao et al., 2019): An LSTM-based multi-stage model without using knowledge.

LM-Based Plan & Write : We replaced the LSTMs used in Plan & Write with BART and GPT-2. The form of data for training is the same as in Yao et al. (2019).

HINT (Guan et al., 2021): A language model-based model that considers the high-level features in the context to improve the coherence.

GPT-2 (Radford et al., 2019) : We applied the public checkpoint of the pre-trained parameters and then fine-tuned with the ROCStory corpus.

Knowledge-enhanced GPT (Guan et al., 2020): A commonsense knowledge pre-trained model with multitask learning.

KGBART (Liu et al., 2021): They incorporated the complex relations of concepts into the model to generate logical and natural sentences.

GRF (Ji et al., 2020): They used dynamic multi-hop reasoning on multi-relational paths to help the pre-trained model generate reasonable text.

Furthermore, to investigate the effect of the component, we derived a variant of our sentence-level model that generates two events by one schema in one iteration in the PG module, named double-event.

5 Evaluation

5.1 Objective Evaluation

We used the following metrics to compare different models: **BLEU** (Papineni et al., 2002) was used to evaluate the n-gram overlap between a generated story and a human-authored story. We experimented with n=1, 2 (B-1, B-2). The metric to evaluate the diversity of generated text is **Distinct** (Li et al., 2016). Distinct-n calculates the ratio of distinct n-grams to all the generated n-grams. We experimented with $n = 4$ (Dist). **Repetition** (Shao et al., 2019) was used to evaluate the redundancy of generated text. Repetition-n shows the percentage of generated stories containing at least one repeated n-gram. We experimented with $n = 4$ (Rept).

5.2 Subjective Evaluation

We conducted a subjective evaluation with Amazon Mechanical Turk (AMT). The annotators were limited to those in the United States who had high school or above equivalent education. We utilized two aspects, **grammaticality** and **coherence**, to analyze the quality of generated stories. When evaluating each aspect, annotators read two stories from different models and then they selected a better one. A special selection *tie* was possible in each aspect in order to cope with cases where the stories are of similar quality. We randomly sampled 168 pairs of stories and assigned 10 annotators to each pair of stories. We used average scores among the annotators. Because the scores of the baselines for the objective evaluation on the WritingPrompts dataset are definitely lower than our model, we tried the subjective evaluation only on the ROCstory dataset.

5.3 Results and Analysis

5.3.1 Results of the Objective Evaluation

The results of the objective evaluation in the ROCstory dataset are shown in Table 2. Our story-level model outperformed the baselines in terms of BLEU and repetition. This shows our story-level model can generate stories more like human-writing, which indicates structural information provided by the schema makes the model easy to catch the relevant information not given by the prediction of the next event.

The right part of Table 2 shows the results on the WritingPrompts dataset. Unlike the ROCstory dataset, the WritingPrompts dataset is a more complex dataset, which contains more dialogue contents as well as descriptions of the environments. We found that our story-level model outperforms the baselines in all metrics. The higher distinct score and lower repetition score of our model in-

[3]The language models are from https://huggingface.co.

Models	ROCstory				WritingPrompts			
	B-1 ↑	**B-2** ↑	**Dist** ↑	**Rept** ↓	**B-1** ↑	**B-2** ↑	**Dist** ↑	**Rept** ↓
Plan&Write	36.14	26.36	68.89	12.28	19.90	7.00	27.20	74.40
LM-Based Plan & Write	31.85	23.70	39.75	50.94	14.61	12.21	26.71	86.80
HINT†	33.40	15.40	69.30	25.30	22.40	8.40	31.30	75.36
GPT-2	36.47	26.95	72.83	33.28	23.98	20.62	35.71	65.40
Knowledge-enhanced GPT	36.57	26.76	**82.23**	18.82	14.94	12.87	49.21	81.50
KGBART	31.48	22.66	40.15	7.00	-	-	-	-
GRF	35.63	25.77	50.38	68.20	22.39	20.99	51.43	79.30
Our story-level model	**38.23**	**27.67**	74.79	**6.71**	**31.36**	**25.37**	**84.75**	**20.90**
Our sentence-level model	36.61	26.68	65.55	39.80	29.22	24.82	74.86	44.70
double-event model	37.63	27.36	69.08	27.50	30.58	24.90	81.82	40.10
Gold story	N/A	N/A	*95.07*	*3.08*	N/A	N/A	*98.04*	*8.70*

Table 2: Results of the objective evaluation on the ROCstory and WritingPrompts datasets. The values in **bold** are the best performance. The results for the gold stories are in *italics*. Compared with the previous work, our story-level model got higher BLEU-1, 2, and Repetition. †: the results from (Guan et al., 2021).

Models	**Coherence**			**Grammaticality**		
Story-level model vs	Win	Tie	Loss	Win	Tie	Loss
Plan&Write	66.67%**	16.19%	17.14%	36.19%**	51.43%	12.38%
GPT-2	75.24%**	16.67%	8.09%	46.67%**	42.38%	10.95%
Knowledge-enhanced GPT	51.90%**	10.95%	37.15%	47.62%**	19.52%	32.86%
KGBART	48.57%**	22.86%	28.57%	43.34%**	33.33%	23.33%
GRF	51.43%**	20.00%	28.57%	37.62%**	36.67%	25.71%
Our sentence-level model	46.20%**	21.90%	31.90%	30.00%	48.10%	21.90%
double-event model	45.72%*	19.52%	34.76%	37.14%	29.05%	33.81%

Table 3: Results of the subjective evaluation. Our story-level model obtained better coherence scores than the baselines while keeping grammatical correctness. The scores marked with * and ** mean our story-level model outperforms the other models significantly with $p < 0.05$ and $p < 0.01$ with t-test, respectively.

dicate that the structured knowledge can guide the model to use it more efficiently to produce diverse stories and suppress duplicate contents compared with the previous knowledge-incorporated models. As the schema is dynamic and contextualized structured knowledge, it provides better necessary information for story generation than fixed knowledge to control the generation of coherent stories. Therefore, the model can ensure that the generated stories are more human-like, even in complex contexts. The higher BLEU scores in Table 2 reflect the power of the schema.

However, we observed that utilizing a story-level schema would reduce the diversity of generated stories, causing our model to perform worse than the Knowledge-enhanced GPT in the ROCstory dataset. We analyze that more information might be contained in the events: One reason might be that this gives stricter constraints to GPT-2, which increases the generation difficulty. These constraints limit the space for the details being able to be added. The other reason might be that GPT-2 needs more cost to balance the quality of the generated sentences and the integrity of information in the events. However, these constraints also control irrelevant content generation, leading to high BLEU scores. Otherwise, GPT-2 has more space to add words to a story, which might cause the story to contain incoherent content or repetition.

5.3.2 Results of the Subjective Evaluation

The results of the subjective evaluation are shown in Table 3. Compared with the Knowledge-enhanced GPT, our story-level model had higher coherence and grammaticality scores. Instead of feeding thousands of knowledge to the model to pre-train it, we used only 60 pieces of knowledge

Cases	current event	schema	generated next event
Case 1	get out of the shower	have a shower go in bathroom clean body ...	**go to the bathroom**
	go to the bathroom	go in bathroom take a shower clean oneself ...	**go to the bathroom**
Case 2	**notice a wallet on the ground**	take wallet from ground go to the police look for the owner ...	look for the owner
	look for the owner	look around for person buy the dog pick up the dog ...	**take the dog to a trainer**

Table 4: Examples of a repeated event and incoherent event generated by the sentence-level model. Case 1 shows a similar schema causing repeated events. Case 2 shows a lack of context causing incoherent events.

for a schema, which shows that the schema is more useful for our model to effectively guide the generation. Compared with KGBART and GRF, our story-level model still had better performance in terms of coherence and grammaticality, which indicates a structured schema can help the model catch more relevant information.

5.3.3 Analysis of Our Different Models

The sentence-level model performed poorly compared with the story-level model in both evaluations, while it provides more schemata. To investigate the reasons, we illustrate two cases in Table 4. First, we observed that, although we update the schema in the sentence-level model in every step, if the current event is similar to the previous one, the knowledge pieces in the previous schema will also be in the updated schema (e.g. *go in bathroom*). Because the schema takes up most of the input space, it has a heavy weight for affecting the events generated in the PG module. Obviously, homogeneous knowledge in the input leads to repeated events to be generated in the plot (Case 1), which will cause generated stories with repetitions, as reflected in the high repetition score in Table 2.

Second, because a sentence-level schema is generated depending only on the current event, the schemata for a whole plot tend to contain a lot of inconsistent knowledge pieces (in Case 2). Using only the current sentence leads to a lack of context and a lack of control. As a result, inconsistent events are generated (the information *dog* is related to *owner*, but not related to *wallet*), which leads to incomprehensible stories.

To reduce the repeated information in the schema and enhance the control from the context, we set the double-event model to compare with. In our double-event model, we generate two events in one iteration by the same schema in the PG module, and then update the schema by the generated events. In this model, although the schema contains less information than in the sentence-level model, it can keep more context when generating the sequential events, and avoid repetition. As shown in Table 2, we can see the two problems in the sentence-level model are alleviated. The double-event model gets better performance than our sentence-level model in all of the objective evaluation metrics.

In contrast, in the story-level model, the subsequent events for a story are generated together, and only a schema is generated depending only on a given event, while it might cause less diversity.

6 Conclusion

We presented a knowledge-based multi-stage model for coherent story generation. A structured knowledge, schema, was applied to navigate the story generation process, which makes the model able to readily absorb and integrate the knowledge not contained in the context to generate coherent content. The results of objective and subjective evaluations of the datasets showed that the proposed method outperforms strong baselines and often produces stories with more coherence and less repetition without harming grammatical correctness. Furthermore, by exploring our different models, we found some limitations in the usage of knowledge in the multi-stage models. We hope our work can give good guidance to future work.

References

Prithviraj Ammanabrolu, Ethan Tien, Wesley Cheung, Zhaochen Luo, William Ma, Lara J. Martin, and Mark O. Riedl. 2020. Story realization: Expanding plot events into sentences. In *AAAI 2020*, pages 7375–7382. AAAI Press.

Richard C Anderson and P David Pearson. 1984. *A schema-theoretic view of basic processes in reading comprehension*. Handbook of reading research, New York.

APA Dictionary. 2022. Schema. https://dictionary.apa.org/schema. Accessed: 2022-07-25.

W. F. Brewer and G. V. Nakamura. 1984. The nature and functions of schemas. 1:119–160.

Khyathi Chandu, Shrimai Prabhumoye, Ruslan Salakhutdinov, and Alan W Black. 2019. "my way of telling a story": Persona based grounded story generation. In *Proceedings of the Second Workshop on Storytelling*, pages 11–21, Florence, Italy. Association for Computational Linguistics.

Angela Fan, Mike Lewis, and Yann Dauphin. 2018. Hierarchical neural story generation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 889–898, Melbourne, Australia. Association for Computational Linguistics.

Angela Fan, Mike Lewis, and Yann Dauphin. 2019. Strategies for structuring story generation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2650–2660, Florence, Italy. Association for Computational Linguistics.

Jian Guan, Fei Huang, Zhihao Zhao, Xiaoyan Zhu, and Minlie Huang. 2020. A knowledge-enhanced pretraining model for commonsense story generation. *Transactions of the Association for Computational Linguistics*, 8:93–108.

Jian Guan, Xiaoxi Mao, Changjie Fan, Zitao Liu, Wenbiao Ding, and Minlie Huang. 2021. Long text generation by modeling sentence-level and discourse-level coherence. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 6379–6393, Online. Association for Computational Linguistics.

Aditya Gupta and Greg Durrett. 2019. Tracking discrete and continuous entity state for process understanding. In *Proceedings of the Third Workshop on Structured Prediction for NLP*, pages 7–12, Minneapolis, Minnesota. Association for Computational Linguistics.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2020. The curious case of neural text degeneration. In *ICLR*. OpenReview.net.

Jena D. Hwang, Chandra Bhagavatula, Ronan Le Bras, Jeff Da, Keisuke Sakaguchi, Antoine Bosselut, and Yejin Choi. 2021. (comet-) atomic 2020: On symbolic and neural commonsense knowledge graphs. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021,*, pages 6384–6392. AAAI Press.

Haozhe Ji, Pei Ke, Shaohan Huang, Furu Wei, Xiaoyan Zhu, and Minlie Huang. 2020. Language generation with multi-hop reasoning on commonsense knowledge graph. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 725–736, Online. Association for Computational Linguistics.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. 2016. A diversity-promoting objective function for neural conversation models. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 110–119, San Diego, California. Association for Computational Linguistics.

Shih-Ting Lin, Nathanael Chambers, and Greg Durrett. 2021. Conditional generation of temporally-ordered event sequences. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 7142–7157, Online. Association for Computational Linguistics.

Ye Liu, Yao Wan, Lifang He, Hao Peng, and Philip S. Yu. 2021. KG-BART: knowledge graph-augmented BART for generative commonsense reasoning. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021*, pages 6418–6425. AAAI Press.

Robert Logan, Nelson F. Liu, Matthew E. Peters, Matt Gardner, and Sameer Singh. 2019. Barack's wife hillary: Using knowledge graphs for fact-aware language modeling. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5962–5971, Florence, Italy. Association for Computational Linguistics.

Nasrin Mostafazadeh, Nathanael Chambers, Xiaodong He, Devi Parikh, Dhruv Batra, Lucy Vanderwende, Pushmeet Kohli, and James Allen. 2016. A corpus and cloze evaluation for deeper understanding of commonsense stories. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 839–849, San Diego,

California. Association for Computational Linguistics.

Nasrin Mostafazadeh, Michael Roth, Annie Louis, Nathanael Chambers, and James Allen. 2017. LSDSem 2017 shared task: The story cloze test. In *Proceedings of the 2nd Workshop on Linking Models of Lexical, Sentential and Discourse-level Semantics*, pages 46–51, Valencia, Spain. Association for Computational Linguistics.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Xiangyu Peng, Siyan Li, Sarah Wiegreffe, and Mark O. Riedl. 2021. Inferring the reader: Guiding automated story generation with commonsense reasoning.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

Hannah Rashkin, Asli Celikyilmaz, Yejin Choi, and Jianfeng Gao. 2020. PlotMachines: Outline-conditioned generation with dynamic plot state tracking. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4274–4295, Online. Association for Computational Linguistics.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.

Mark Sadoski, Allan Paivio, and Ernest T. Goetz. 1991. Commentary: A critique of schema theory in reading and a dual coding alternative. *Reading Research Quarterly*, 26(4):463–484.

Keisuke Sakaguchi, Chandra Bhagavatula, Ronan Le Bras, Niket Tandon, Peter Clark, and Yejin Choi. 2021. proScript: Partially ordered scripts generation. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 2138–2149, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Zhihong Shao, Minlie Huang, Jiangtao Wen, Wenfei Xu, and Xiaoyan Zhu. 2019. Long and diverse text generation with planning-based hierarchical variational model. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3257–3268, Hong Kong, China. Association for Computational Linguistics.

Jian Wang, Junhao Liu, Wei Bi, Xiaojiang Liu, Kejing He, Ruifeng Xu, and Min Yang. 2020. Improving knowledge-aware dialogue generation via knowledge base question answering. In *AAAI 2020*, pages 9169–9176. AAAI Press.

Wenhan Xiong, Jingfei Du, William Yang Wang, and Veselin Stoyanov. 2020. Pretrained encyclopedia: Weakly supervised knowledge-pretrained language model. In *ICLR 2020*. OpenReview.net.

Jingjing Xu, Xuancheng Ren, Yi Zhang, Qi Zeng, Xiaoyan Cai, and Xu Sun. 2018. A skeleton-based model for promoting coherence among sentences in narrative story generation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4306–4315, Brussels, Belgium. Association for Computational Linguistics.

An Yang, Quan Wang, Jing Liu, Kai Liu, Yajuan Lyu, Hua Wu, Qiaoqiao She, and Sujian Li. 2019a. Enhancing pre-trained language representations with rich knowledge for machine reading comprehension. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2346–2357, Florence, Italy. Association for Computational Linguistics.

Pengcheng Yang, Lei Li, Fuli Luo, Tianyu Liu, and Xu Sun. 2019b. Enhancing topic-to-essay generation with external commonsense knowledge. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2002–2012, Florence, Italy. Association for Computational Linguistics.

Lili Yao, Nanyun Peng, Ralph M. Weischedel, Kevin Knight, Dongyan Zhao, and Rui Yan. 2019. Plan-and-write: Towards better automatic storytelling. In *AAAI2019*, pages 7378–7385. AAAI Press.

Fangzhou Zhai, Vera Demberg, Pavel Shkadzko, Wei Shi, and Asad Sayeed. 2019. A hybrid model for globally coherent story generation. In *Proceedings of the Second Workshop on Storytelling*, pages 34–45, Florence, Italy. Association for Computational Linguistics.

Chujie Zheng and Minlie Huang. 2021. Exploring prompt-based few-shot learning for grounded dialog generation.

Guo Zhipeng, Xiaoyuan Yi, Maosong Sun, Wenhao Li, Cheng Yang, Jiannan Liang, Huimin Chen, Yuhui Zhang, and Ruoyu Li. 2019. Jiuge: A human-machine collaborative Chinese classical poetry generation system. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 25–30, Florence, Italy. Association for Computational Linguistics.