Conveying the Predicted Future to Users: A Case Study of Story Plot Prediction

Chieh-Yang Huang,[1] **Saniya Naphade,**[2*] **Kavya Laalasa Karanam,**[3*]
Ting-Hao 'Kenneth' Huang[1]

[1] Pennsylvania State University, University Park, PA, USA. {chiehyang,txh710}@psu.edu
[2] GumGum Inc., Los Angeles, CA, USA. saniya.naphade@gmail.com
[3] Intel Corporation, Santa Clara, CA, USA. laalasa.kar@gmail.com

Abstract

Creative writing is hard: Novelists struggle with writer's block daily. While automatic story generation has advanced recently, it is treated as a "toy task" for advancing artificial intelligence rather than helping people. In this paper, we create a system that produces a short description that narrates a predicted plot using existing story generation approaches. Our goal is to assist writers in crafting a consistent and compelling story arc. We conducted experiments on Amazon Mechanical Turk (AMT) to examine the quality of the generated story plots in terms of consistency and *storiability*. The results show that short descriptions produced by our frame-enhanced GPT-2 (FGPT-2) were rated as the most consistent and *storiable* among all models; FGPT-2's outputs even beat some random story snippets written by humans. Next, we conducted a preliminary user study using a story continuation task where AMT workers were given access to machine-generated story plots and asked to write a follow-up story. FGPT-2 could positively affect the writing process, though people favor other baselines more. Our study shed some light on the possibilities of future creative writing support systems beyond the scope of completing sentences. Our code is available at: https://github.com/appleternity/Story-Plot-Generation.

Introduction

Storytelling is an important human activity. People engage in storytelling to communicate, teach, entertain, establish identity, or relate to each other in meaningful ways. However, creative writing is known to be a cognitively demanding task, and writers struggle with writer's block daily. Researchers and the industry have created a series of techniques that support human writing. Many techniques focus on lower-level language support, such as auto-completion, grammar checking, or typo detection, and these have proven helpful and are widely used. On the other hand, the techniques aiming to provide higher-level guidance, such as story generation, have long been treated only as in-the-lab artificial intelligence tasks. Automatic story generation, for example, was primarily developed and tested using toy datasets composed of stories that are *(i)* extremely short (for example, containing five sentences, such as ROCStories (Mostafazadeh et al. 2016)), *(ii)* written under artificial constraints to make it easier for machines to learn (*e.g.*, GLUCOSE (Mostafazadeh et al. 2020)), or *(iii)* based on the assumption of a story starter prompt (*e.g.*, WritingPrompt (Fan, Lewis, and Dauphin 2018)). However, real-world writers compose novels with over 10,000 words, work with blank pages with few constraints, and can get stuck anywhere in the middle of a draft. As the models trained on toy datasets inevitably generate stories inheriting the data's characteristics, it is unclear how well modern story generation models can be used to support writers in practice.

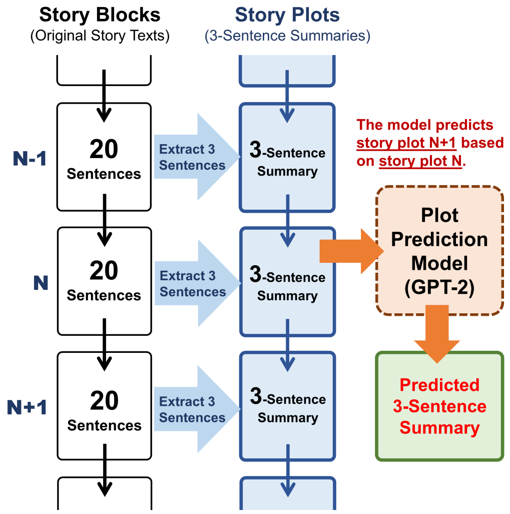


Figure 1: We view a long novel as a sequence of fixed-sized story blocks. The goal of the proposed task is to consider the previous story block (*i.e.*, B_n) and generate a short description for the next story block (*i.e.*, B_{n+1}). We define a short description as a three-sentence summary of a story block.

In this paper, we aim to support creative writing in practical terms. We view a long novel as a sequence of fixed-sized story blocks (*e.g.*, 20 sentences). The goal is to generate a short description that narrates future story plots for the next story block (*i.e.*, B_{n+1}) using the previous story block (*i.e.*, B_n). We define a story plot as a short summary over a story block that illustrates the key follow-up idea instead of the detailed full text. Three existing story generation models, Fusion-based seq2seq (Fan, Lewis, and Dauphin 2018),

[*] These authors contributed equally.

Plan-and-Write (Yao et al. 2019), and GPT-2 (Radford et al. 2019) enhanced with semantic frame representation (Huang and Huang 2021), are adapted to predict the follow-up story plot given the context of the previous story block.

We first conduct a quality assessment study on Amazon Mechanical Turk (AMT) to measure the quality of the machine-generated story plots. In this study, crowd workers are recruited to *(i)* read a previous story block and six follow-up story plots, and *(ii)* rank the quality in terms of consistency and storiability (Roemmele 2021). The experiment shows that story plots generated by our frame-enhanced GPT-2 are more consistent than randomly selected plots written by humans and are competitive with them in the sense of storiability. The result suggests that human-written plots are still strong baselines, especially the ground truth, but frame-enhanced GPT-2 is capable of generating consistent and storiable story plots to a certain level.

We further conduct a writing task study on AMT to understand how much humans can benefit from machine-generated plots. In this study, crowd workers are asked to develop a 100-word follow-up story given the previous story block and the four follow-up story plots as hints. After finishing the writing task, we collect crowd workers' self-reported judgments on four aspects: degree of inspiration, helpfulness, readability, and creativity. The result shows that frame-enhanced GPT-2 produces output that is less inspiring when compared to strong baselines such as ground truth and GPT-3. However, analyses of the written stories also suggest that, despite being less favored by humans, frame-enhanced GPT-2 still has a positive influence on the written story draft. This finding also echoes Roemmele (2021)'s inspiration-through-observation paradigm: Human writing can still be improved even with less storiable machine-generated texts.

Related Work

Our work is mainly related to *(i)* supporting creative writing and *(ii)* story generation.

Supporting Creative Writing

Prior research has supported creative writing in different ways. InkWell mimics a specified writer's personality traits and revises the draft to provide stylistic variations (Gabriel, Chen, and Nichols 2015). Metaphoria generates metaphorical connections according to the user's input to help create metaphors (Gero and Chilton 2019). Creative Help generates a follow-up sentence as a suggestion for creative writing using a recurrent neural network (Roemmele and Gordon 2015). Heteroglossia collects story plot ideas using a crowdsourcing approach to support writers in continuing the story when stuck due to writer's block (Huang, Huang, and Huang 2020). Scheherazade is built for interactive narrative generation with a crowd-powered system to collect narrative examples (Li and Riedl 2015). Clark et al. (2018) explore the process of machine-in-the-loop creative writing and find that machine-generated suggestions should achieve a balance between coherency and surprise. Roemmele (2021) studies the inspiration-through-observation paradigm and finds that people produce appealing sentences when observing the generated examples. Compass identifies and fills in unnoticed missing information in stories (Mori et al. 2022). Padmakumar and He (2022) build a machine-in-the-loop system to rewrite a span of text or fill in sentences between two pieces of text when requested.

Recently, large language models (LLMs) have shown incredible power in text continuation, rewriting, few-shot learning, and so on. Many researchers have explored how LLMs can be used to support creativity. Storium fine-tunes GPT-2 to consider complicated contextual information (intro, character, and so on) to generate a few follow-up sentences to continue the story (Akoury et al. 2020). Story Centaur provides an interface where users can provide few-shot learning examples to teach LLMs new functions (Swanson et al. 2021). CoPoet, a collaborative poetry writing system, allows users to control an LLM by specifying the attributes of the desired text (Chakrabarty, Padmakumar, and He 2022). TaleBrush allows users to control a protagonist's fortune through a line sketching interaction, and the user-specified fortune curve is used to guide an LLM's story generation process (Chung et al. 2022). CoAuthor supports writing by providing a sentence to continue the given draft by GPT-3 (Lee, Liang, and Yang 2022). Sparks inspires scientific writing by using LLMs to generate sentences that could spark users' ideas (Gero, Liu, and Chilton 2022). Wordcraft allows users to interact with LLMs through a chatbot interface (Ippolito et al. 2022). Dramatron, built with an LLM with prompt-chaining mechanism, could write theatre scripts and screenplays together with users (Mirowski et al. 2022). Unlike most of the prior works, where generated sentences are ready to use in the story, our work aims to generate a short summary for the follow-up story and expects users to develop exact story content manually.

Story Generation

Traditional story generation focuses on producing logically coherent stories using planning or reasoning-based approaches (Riedl and Young 2010; Li et al. 2013). Recently, neural story generation models (Peng et al. 2018; Fan, Lewis, and Dauphin 2018) and pre-trained models (Radford et al. 2019; Keskar et al. 2019) have been used for story generation in an end-to-end manner. However, these models still suffer from the issue of generating repetitive and insufficiently diverse stories (See et al. 2019). To further enhance the coherence among sentences and events, researchers design a variety of intermediate representations to guide the story generation process, including event triplets (Martin et al. 2018), keyword storylines (Yao et al. 2019), critical phrases (Xu et al. 2018), action plans with semantic role labeling (SRL) (Fan, Lewis, and Dauphin 2019), content planning (keyphrase and sentence-level position) (Hua and Wang 2020), and plot structure based on SRL (Goldfarb-Tarrant et al. 2020). However, most of these work on short stories, such as WritingPrompt (Fan, Lewis, and Dauphin 2018), ROCStories (Mostafazadeh et al. 2016), or WikiPlots (Bamman, O'Connor, and Smith 2013). Unlike real-world novels– which usually have more than 10,000 words– the stories from these datasets often end up with under 1,000 or even 100 words.

Plot Prediction

We follow Huang and Huang to split a full story into a sequence of story blocks and each story block contains a fixed number of sentences. Note that the size of the story block can vary to fulfill different purposes. Large story blocks (200 sentences or beyond) can capture the high-level ideas among chapters; whereas small story blocks (five or ten sentences) can be used to model the event relationships in a near future. In this paper, we focus on medium size story blocks (20 sentences).

Next, we define **a story plot** as **a three-sentence summary** of a huge story block. The plot prediction task, thus, is defined as using the story plot in story block *n* to predict the story plot in story block *n+1*. In this section, we first describe how we collect the story plots using the extractive summarization model; and then detail how we adapt the three existing story generation models to our problem.

Collecting Story Plots

To generate such a summary for every story block, we use Matchsum (Zhong et al. 2020), an extractive summarization model. We train the Matchsum model on the Booksum dataset (Kryściński et al. 2021) where each paragraph is paired with a one- or two-sentence summary. To ensure the training instances from Booksum are similar to those in our story block setup (20 sentences), only 20,709 paragraphs with more than 10 sentences are kept for training. The fine-tuned Matchsum model is then applied to our Bookcorpus dataset to generate the story plot for all the 900k story blocks. We randomly select 1k instances as the validation set in order to observe if the model converges or not in the training process.

Story Plot Generation Models

Here, we adapt three existing models to our story plot generation task: (*i*) Fusion-based Seq2Seq (Fan, Lewis, and Dauphin 2018), (*ii*) Plan-and-Write (Yao et al. 2019), and (*iii*) GPT-2 (Radford et al. 2019) guided by semantic frame representation (Huang and Huang 2021).

Fusion-Based Seq2seq. The fusion-based mechanism (Fan, Lewis, and Dauphin 2018) is a hierarchical model where a seq2seq model is trained on top of a pre-trained seq2seq model. The underlying Convolutional Seq2Seq model is trained on a premise or prompts, the plot of story block *n* in our case. The fusion model, another convolutional seq2seq model is then trained on top of the previous seq2seq model to encourage the model to focus on the link between the prompt and the generated story, making it easier to generate consistent stories and reducing the tendency to drift off-topic. Given the prompt, the model generates one of the possible directions story block *n+1* in which the story could progress further.

We tokenize the plots using NLTK, turn words that appear less than 10 times in the training set to `<UNK>`, and follow the paper's default hyper-parameters to train the model (Fan, Lewis, and Dauphin 2018). The base seq2seq model is trained for 20 epochs and then used to train the fusion model which takes 15 epochs. We use top-k sampling to generate story plots with k = 100, temperature = 0.8, and unknown token penalty = 100. Plot lengths are limited to 31 to 71 tokens as the average length in the training set is 51.08.

Plan-and-Write (P&W). Plan-and-Write (Yao et al. 2019) makes use of static planning which generates storylines as a standard intermediate representation to create coherent and diverse stories. Storylines are depicted as a sequence of important words that estimate structures for a real story plot. The P&W model takes a prompt as the input to (*i*) first plan the storylines and (*ii*) then generate the whole story. Following Yao et al. (2019)'s setup, we apply RAKE algorithm (Rose et al. 2010) to extract keywords from the plot of story block *n+1* to form the storylines.

We tokenize the plots using NLTK and turn words that appear less than 10 times in the training set to `<UNK>`. The storyline generation model is based on a 3-layer LSTM with embedding size = 300, hidden size = 300; and the plot generation model is based on a 5-layer LSTM with embedding size = 512, hidden size = 512; We use Adam (Kingma and Ba 2015) as the optimizer to train the model. For the rest of the hyper-parameters, we follow the setting in the original paper. The storyline model is trained for 100 epochs and the plot generation model is trained for 40 epochs. We use temperature sampling to generate storylines and the final story plots with temperature = 0.8. Plan-and-Write does not handle unknown tokens so we add our implementation by setting the sampling probability of `<UNK>` to zero. Again, plot lengths are limited to 31 to 71 tokens. After obtaining the generated story plots, to remove the artifact, we further apply the Treebank detokenizer (Bird, Klein, and Loper 2009) to remove extra spaces and Truecase (Lita et al. 2003) to capitalize necessary letters.

Frame-enhanced GPT-2 (FGPT-2). Semantic frame representations (Huang and Huang 2021) encode high-level semantic units into vectors using their corresponding importance, which has been shown to be effective for representing a longer context. We take advantage of the semantic frame representation to encode longer contextual information and Huang and Huang (2021)'s semantic frame forecast model to generate the guidance of the follow-up story block. Note that the predicted frame representation contains semantic units that are expected to happen in the next story block which serves as a **goal-setting** function in the writing process theory (Flower and Hayes 1981). We build a sequence-to-sequence model using the pre-trained GPT-2 model where two GPT-2 models are used for the encoder and decoder respectively. The frame representation of story block *n* and the predicted frame representation of story block *n+1* is passed through a linear layer to fit the GPT-2's word embedding dimension and inputted into GPT-2 encoder as two words. The encoder input can be described as: $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_n, \mathbf{f}_n, \hat{\mathbf{f}}_{n+1}]$ where $\mathbf{x}_i$ is the i-th word in the story plot of story block *n*, $\mathbf{f}_n$ is the adapted frame representation of story block *n*, and $\hat{\mathbf{f}}_{n+1}$ is the LGBM's prediction of the frame representation for story block *n+1*. We added the `START` and `<PAD>` tokens to the model to enable the sequence-to-sequence training framework. The sequence

	BLEU-4	METEOR	ROUGE-L	ST
Rand-History	.0012	.0868	.1663	.6101
Rand-Future	.0008	.0881	.1661	**.6106**
Fusion-Seq	.0002	.0741	.1609	.5700
P&W	.0005	.0814	.1686	.5836
FGPT-2	**.0015**	**.0919**	**.1744**	.5905

Table 1: Automatic evaluation metrics for the plot prediction task using Ground-Truth as the reference. Out of the five models, FGPT-2 can generate story plots that are more similar to Ground-Truth.

$\downarrow$	GT	RH	RF	Fusion	P&W	FGPT-2
Mean	3.091	3.586	3.528	3.733	3.741	3.321
P-values for T-test						
GT	-	<0.001	<0.001	<0.001	<0.001	0.003**
RH	-	-	0.437	0.054	0.039*	<0.001
RF	-	-	-	0.006**	0.004**	0.005*
Fusion	-	-	-	-	0.915	<0.001
P&W	-	-	-	-	-	<0.001

Table 2: Consistency ranking result shows that Ground-Truth $\ll$ FGPT-2 $\ll$ Random-Future $<$ Random-History $\ll$ Fusion-Seq $<$ P&W, where $\ll$ stands for "significantly better". (n=1000)

$\downarrow$	GT	RH	RF	Fusion	P&W	FGPT-2
Mean	3.178	3.402	3.452	3.756	3.748	3.464
P-values for T-test						
GT	-	0.003**	<0.001	<0.001	<0.001	<0.001
RH	-	-	0.518	<0.001	<0.001	0.414
RF	-	-	-	<0.001	<0.001	0.877
Fusion	-	-	-	-	0.915	<0.001
P&W	-	-	-	-	-	<0.001

Table 3: Storiability ranking result shows that Ground-Truth $\ll$ Random-History $<$ Random-Future $<$ FGPT-2 $\ll$ P&W $<$ Fusion-Seq, where $\ll$ stands for "significantly better". (n=1000)

[<START>, $y_1, y_2, \cdots, y_n$,<|endoftext|>], where y_i is the i-th word in the story plot of story block *n+1*, is used as the target to train the decoder.

The model is built using HuggingFace's GPT-2 implementationand is trained using AdamW optimizer (Loshchilov and Hutter 2019) with initial learning rate = $3e-4$, weight decay factor = $1e-4$. The model is trained for 800,000 steps with batch size = 32. Top-k sampling is used for generating story plots with top-k = 100, temperature = 0.8, and repetition penalty = 3.0. Plot lengths are limited to 36 to 76 tokens as the average plot length using GPT-2's tokenizer is 56.

Quality Assessment

In this study, we examine the quality of plot generation.

Plot Prediction Assessment

In this plot prediction assessment task, the goal is to measure the quality of the automatically generated plots by the three models described in the Plot Prediction Section. We take the human-written plots extracted from the real book as the baselines. A total of six different models are compared.

- **Ground-Truth (GT).** The gold standard story plot of the story block *n+1*. We expect Ground-Truth to be the upper bound. Note that the story plot is obtained by applying the Matchsum model to extract a three-sentence summary on the target story block.
- **Random-Future (RF).** The story plot of a randomly selected story block *n+u*, where $5 \leq u \leq 15$. Random-Future is expected to be a strong baseline as it is a follow-up story plot that happened later.
- **Random-History (RH).** The story plot of a randomly selected story block *n-t*, where $10 \leq t \leq 20$. We expect Random-History to be a slightly weaker baseline as it is something that happened.
- **Fusion-Seq.** The fusion-based Seq2seq model described in the Plot Prediction section.
- **P&W.** The plan-and-write model described in the Plot Prediction section.
- **FGPT-2.** The sequence-to-sequence GPT-2 model guided by frame representations as described in the Plot Prediction section.

The evaluation data is built from the Huang and Huang's Bookcorpus testing set (Huang and Huang 2021) where 958 qualified books are collected. We randomly sample one story block as the target story block n from each book to create a total of 958 testing instances. In the following sections, we describe how we assess the story plot quality using both automatic evaluation metrics and human judgments.

Automatic Evaluation

We evaluate the five generated plots by using the Ground-Truth as the reference. NLG-eval package (Sharma et al. 2017) is used and four common metrics, BLEU-4, METEOR, ROUGE-L, and SkipThought Cosine Similarity (ST), are reported in Table 1. The results show that FGPT-2 outperforms other models in the metrics based on token-overlapping, BLEU-4, METEOR, and ROUGE-L. In the semantic-based metric, ST, the human-written plots, Random-History and Random-Future, still perform better. However, prior works have shown that automatic evaluation metrics, especially token-overlapping metrics, are not entirely suitable for evaluating the story generation domain as stories that differed from the target can still be good stories (Hsu et al. 2019). Therefore, we also conduct a human evaluation to measure the quality.

Human Evaluation

We conduct a human evaluation on AMT to evaluate two aspects of the story plot quality, consistency and storiability. In this task, workers are instructed to read a story snippet (20 sentences) along with six follow-up story plots. We then ask workers to rank the story plots according to consistency and storiability. Consistency assesses whether the given story plot makes sense in its context (story snippet); storiability measures whether readers would be curious to read the complete story developed from the given story plot (Roemmele 2021). Note that we only ask one single question in a HIT so that workers would not get confused. To make sure workers spend enough time reading the story snippet and the story plots, we add a 30-second submission lock to the interface. To alleviate the negative effect and bias caused by the decoding process (such as an unfinished sentence) and control the reading time, we apply the following three rules to create the evaluation set: (*i*) all the six story plots have to be within 25-65 words; (*ii*) the story block *n* has to be within 150-300 words; (*iii*) the story block *n* and *n+1* do not cross chapters. Out of the 247 qualified instances, we randomly select 200 instances for evaluation. For each instance, we collect five assignments which result in a total of 1,000 rankings per aspect. Given that each HIT contains around 500 words to read, we estimate the task to take around 2 minutes and thus we pay $0.33 per assignment.

The consistency ranking results in Table 2 show that Ground-Truth $\ll$ FGPT-2 $\ll$ Random-Future $<$ Random-History $\ll$ Fusion-Seq $<$ P&W, where $\ll$ stands for "significantly better". Surprisingly, FGPT-2 is ranked to be more consistent than the two human-written story plots, Random-Future and Random-History. The other two story plot generation models, Fusion-Seq and P&W, are believed to have lower consistency. Table 3 shows the storibility rankings: Ground-Truth $\ll$ Random-History $<$ Random-Future $<$ FGPT-2 $\ll$ P&W $<$ Fusion-Seq. Again, Ground-Truth serves as the upper bound and is considered to be the most storiable one. Although FGPT-2 does not outperform the two human-written story plot baselines here, it achieves the same level of storiability as them. We thus conclude that FGPT-2 is a good choice for plot generation as it can produce consistent and storiable story plots. Table 4 shows an example output of the six models for comparison.

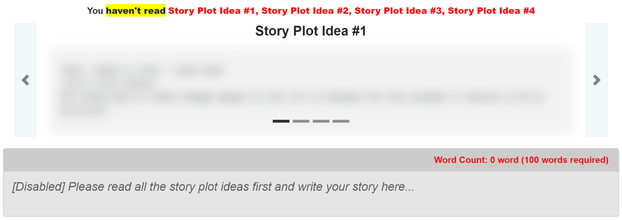
Human Evaluation through a Writing Task

The quality assessment experiment suggests that FGPT-2 could generate story plots that achieve the level of random human-written baselines (Random-Future and Random-History) in terms of consistency and storability. To understand how the generated story plots influence people's story writing, we conduct a preliminary study on AMT using a story continuation task. In this section, we first describe the study protocol and discuss the result.

Baselines

Since writing task is difficult in general, to prevent adding too much workload to workers, we only compare four different models.

- **Ground-Truth (GT).** As described above.
- **Random-Future (RF).** As described above.
- **FGPT-2.** As described above.



You **haven't read** Story Plot Idea #1, Story Plot Idea #2, Story Plot Idea #3, Story Plot Idea #4

Story Plot Idea #1

Word Count: 0 word (100 words required)

[Disabled] Please read all the story plot ideas first and write your story here...

Figure 2: A section of the interface used for the story continuation task. We blur the story plots to prevent workers from copying the exact story plots. Workers would need to click on the story plots to unblur them.

- **GPT-3.** We use OpenAI's GPT-3 API (Brown et al. 2020) to generate follow-up story plots with the prompt, "Given the story snippet: [story] Describe a follow-up story arc within 30 words". We fill in the full story block *n* into [story]. Parameters used was model = text-davinci-002, temperature = 0.95, max-tokens = 76, top-p = 0.95, frequency-penalty = 0.5, presence-penalty = 0.5, and best-of = 5.

Study Protocol

In this story continuation task, workers are asked to finish three steps: (*i*) reading through a story snippet; (*ii*) writing a 100-word follow-up story to continue the given story snippet with access to story plots; and (*iii*) answering questions about their experience.

We first show a 20-sentence story snippet (story block *n*) and ask workers to carefully read through it in step (*i*). In step (*ii*), four story plots generated by four different methods are provided. Note that the order of the four story plots is randomized. After finishing reading all the story plots, workers are instructed to write a 100-word story to continue the given story snippet. As shown in Figure 2, to prevent workers from simply copying-and-pasting or typing in the exact story plot ideas, we blur the four story plots. Workers need to click on the story plots to see the exact texts (the texts would get blurred when the mouse leaves.) After finishing the writing task, we ask workers to fill-up a short questionnaire regarding their experience with the story plots. In the questionnaire, we ask seven questions to measure four different aspects:

1. **Inspiringness.** All story plots that inspire the writing (Q1). Multiple selections are allowed.
2. **Helpfulness.** The most/least helpful story plot (Q2, Q3).
3. **Readability.** The easiest/hardest story plot to comprehend (Q4, Q5).
4. **Creativity.** The most/least creative story plot (Q6, Q7).

We add a 180-second submission lock (workers are allowed to submit after 180 seconds) to the interface to ensure workers spend enough time working on the task. Each task is estimated to be finished within six to nine minutes. Aiming at paying $10 dollars per hour, we set the payment for each task to $1.5 dollars.

Story Snippet. He looked away, blinking back a tear. The bunker was more centered than Josue had realized, protected on all sides by as much of the manor as possible. He liked that. Anywhere he would walk within the new manor, he would be close to his father's final resting place. John looked down at the square section. ”We could rebuild the bunker, if you'd like.” Josue was sure, even if they rebuilt it, he would never find the wherewithal to use it. He couldn't think of a more fitting memorial to his father than to leave it the way it was. ”Please, don't change a thing.” He looked out at the perimeter of the compound. They had almost finished the walls. ”Those will protect us.” ”They will hold for the immediate need.” John led Josue through the compound to the eastern hillside overlooking the manor. The shimmer of the obfuscator remained above them. A separate wall surrounded a new plot. He felt a sense of reverence as the Elder led him into the enclosure. In the last rays of the day sun, symmetric monuments reflected pink, planted in neat rows and columns along the hillside. ”We took the liberty of adding a memorial garden.” John walked to the top of the hill, where a solitary statue stood apart from the others, reminding Josue of a General reviewing his troops.

Ground-Truth. Josue wiped his damp cheek. A fire burned in his breast at the very sight of the name. He died doing what he could to protect those he loved.”

Random-History. Josue tried to remember if he had been on Omri property. Keep away from the mine for a while. His father caught himself and put on a tired smile.

Random-Future. Master Hector raised an eyebrow. Josue pushed the thoughts of Timeos out of his mind. Hector pulled a longpole from the rack and threw it to Josue.

Fusion-Seq. he shouted. The mob rushed up to the door, and with a sickening clang, the doors began to snap open. With a flash, it was the assassin, the assassin.

P&W. John thought it was a best thing. Bill said, ”I have to go home and start a new life.” Tom asked, looking at the sign.

FGPT-2. John walked back to the edge of the building and looked down. He could see the end of the main staircase on the other side of the wall. It was lined with columns, each leading up to the roof.

Table 4: Example of the story snippet and the six follow-up story plots.

Aspect		**GT**	**RF**	**FGPT-2**	**GPT-3**
Inspiringness ↑		0.294	0.294	0.176	**0.647**
Helpfulness	**Most ↑**	0.235	0.353	0.059	0.353
	Least ↓	0.000	0.294	0.294	0.412
	Overall ↑	**0.235**	0.059	−0.235	−0.059
Readability	**Easiest ↑**	0.353	0.235	0.176	0.235
	Hardest ↓	0.294	0.059	0.471	0.176
	Overall ↑	0.059	**0.176**	−0.294	0.059
Creativity	**Most ↑**	0.353	0.176	0.000	0.471
	Least ↓	0.176	0.294	0.353	0.176
	Overall ↑	0.176	−0.118	−0.353	**0.294**

Table 5: Questionnaire results from the story continuation task. GPT3 is rated the most inspiring one.

	GT	**RF**	**FGPT-2**	**GPT-3**	**Random**
Similarity	0.816	0.795	0.795	0.840	0.787

Table 6: Semantic similarity between the story plot idea and the written story.

As a preliminary study, we only test on five instances. We simply take the first five instances from the human evaluation of the quality assessment experiment. For each instance, we collect five assignments, resulting in a total of 25 stories along with 25 questionnaires.

Questionnaire Result

After obtaining all 25 assignments, we first check out all the written stories. Despite adding a 180-second submission lock and the blurring function, there are a lot of spamming submissions, such as irrelevant texts, random online stories, and random keystrokes. We read through all the written stories and manually remove the spamming assignments, resulting in a total of 17 assignments remaining.

Table 5 shows the questionnaire results. For inspiringness, as workers are asked to select **all** story plots that inspired their written story, we report the percentage of a model being considered inspiring over the 17 assignments. GPT-3 inspires more than half of the stories (0.647) while FGPT-2 (0.176) is the least inspiring one.

For helpfulness, readability, and creativity, we report the percentage of a model being selected as the most/least helpful, the easiest/hardest readable, and the most/least creative one. The overall score is computed by Most − Least (or Easiest − Hardest). Ground-Truth, Random-Future, and GPT-3 are considered the most helpful, the easiest readable, and the most creative, respectively. We notice that people do not favor FGPT-2 in all three aspects when compared to other baselines. However, FGPT-2 still inspires 17.6% of the stories. Such phenomenon echoes Roemmele's finding from the inspiration-through-observation paradigm, where human writing would be enhanced by machine-generated texts even though the machine-generated texts are less storible.

We also notice that although GPT-3 gets the highest “most helpful” votes, it also gets the highest “least helpful” votes. This would require more analysis to understand why.

How do story plots affect story writing?

The questionnaire serves as self-reported results. To understand how the story plots influence story writing, we analyze

	Story Coverage		Plot Coverage	
	Mean	**CI**	**Mean**	**CI**
GT	0.198	[0.163, 0.233]	0.530	[0.473, 0.587]
RF	0.193	[0.164, 0.222]	0.536	[0.475, 0.598]
FGPT-2	0.163	[0.145, 0.182]	0.484	[0.429, 0.539]
GPT-3	0.170	[0.149, 0.190]	0.498	[0.441, 0.555]
Random	0.151	[0.149, 0.153]	0.450	[0.445, 0.455]

Table 7: GT and RF contribute more in token level.

the relationship between the story plots and the follow-up story written by workers. Note that each HIT assignment comes with one human-written follow-up story and four machine-generated story plots. Using this data, we measure two aspects: (*i*) semantic similarity and (*ii*) token overlap.

Semantic Similarity. We encode the follow-up stories and the story plots using Sentence-BERT (Reimers and Gurevych 2019)[1]. Cosine similarity is then used to compute the semantic similarity between a follow-up story and a story plot. To get a better sense of this cosine similarity value, we also include a Random baseline for comparison. The Random baseline reports the semantic similarity between the follow-up stories and a set of randomly sampled 40-word paragraphs. We choose 40 words to match the length of the story plots. These paragraphs are randomly selected from NLTK's Gutenberg corpus. We start with 3,400 random paragraphs (two sentences) and remove those shorter than 30 words or longer than 50 words. A total of 1,113 valid short paragraphs are then included in this random set.

As shown in Table 6, story plots generated by GPT-3 have the highest semantic similarity (0.840) with the human-written follow-up stories, suggesting that workers do get inspiration from GPT-3. Ground-Truth (0.816), Random-Future (0.795), and FGPT-2 (0.795) are slightly lower. However, all of the models are higher than the Random baseline suggesting that workers do get inspired more or less, which again echos the inspiration-through-observation paradigm (Roemmele 2021).

Token Overlap. To understand the effect in the token level, we use awesome-align (Dou and Neubig 2021) to get the token alignment between the follow-up story and the story plots. Awesome-align computes the similarity over tokens' contextual embeddings to assign alignments. Compared to the exact token overlap, awesome-align provides a soft overlap where semantically similar words can be identified. Upon getting the alignment, we compute two scores.

1. **Story Coverage**: the percentage of the follow-up stories that can be found in the story plots. This also means the amount of information contributed by the story plots.
2. **Plot Coverage**: the percentage of the story plots that can be found in the follow-up stories. This also indicates the helpfulness of the story plots.

Note that when computing the coverage score, we exclude punctuation and stop words. Again, we add a Random baseline to help us interpret the scores. As shown in Table 7, Ground-Truth and Random-Future help more at the token level. This is probably because they provide wording and terms more useful in the context. Although FGPT-2 is less helpful compared to other strong baselines, it still somewhat affects workers' writing.

The difference between semantic helpfulness and token-level helpfulness probably causes workers to vote GPT-3 as the most helpful but also the least helpful.

Discussion

The preliminary experiment on the story continuation task somewhat suggests that the proposed FGPT-2 approach only provides limited help in real writing tasks. We identify a few possible reasons.

Limitations of our story plot formulation. To provide support in long stories, we extract only three sentences from a story block to form a story plot. We expect story plots to capture all the essential information and represent the story block, but much of the information is indeed missing. This is probably caused by the use of the extractive summarization method, which essentially selects a few sentences from the story block. Therefore, if we would like to include all the essential information, the story would need to contain a few very informative sentences. This is not the case for stories. In the future, we would explore other ways to condense information for a story block.

The study on the short story writing task might not capture the difficulty of novel writing. The proposed method is built for stories, such as novels, that are long enough to break recent story generation models. However, conducting experiments with people on such long texts is hard. To evaluate the system in the desired context, we will conduct a formal user study in the future.

Conclusion

In this paper, we generate short future story plots as follow-up story ideas to help writers continue their story. Results on AMT confirm that using FGPT-2 to serve as the plot prediction model yields plots that are significantly more consistent than the ones generated by the two human-written baselines, Random-History and Random-Future. When comparing the storiability, how appealing a story plot is for readers, FGPT-2 is competitive to the two random baselines. A preliminary human study with a story continuation task suggests that FGPT-2 could positively affect story writing but there are still difficulties to overcome. In the future, we will (*i*) explore better ways to build story plots and (*ii*) integrate the proposed function to a real editor (*e.g.*, Google Docs) and conduct studies to measure whether writers can get benefits for fiction writing in practice.

Acknowledgments

The authors extend their heartfelt appreciation to the late Dr. Arzoo Katiyar for her unwavering support and insightful suggestions throughout the course of this project. The authors also acknowledge the valuable contributions of the

[1] `sentence-transformers/sentence-t5-base`

anonymous reviewers, who provided constructive feedback to enhance the quality of this work.

References

Akoury, N.; Wang, S.; Whiting, J.; Hood, S.; Peng, N.; and Iyyer, M. 2020. STORIUM: A Dataset and Evaluation Platform for Machine-in-the-Loop Story Generation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 6470–6484. Online: Association for Computational Linguistics.

Bamman, D.; O'Connor, B.; and Smith, N. A. 2013. Learning Latent Personas of Film Characters. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 352–361. Sofia, Bulgaria: Association for Computational Linguistics.

Bird, S.; Klein, E.; and Loper, E. 2009. *Natural Language Processing with Python*. O'Reilly Media.

Brown, T.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J. D.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; Agarwal, S.; Herbert-Voss, A.; Krueger, G.; Henighan, T.; Child, R.; Ramesh, A.; Ziegler, D.; Wu, J.; Winter, C.; Hesse, C.; Chen, M.; Sigler, E.; Litwin, M.; Gray, S.; Chess, B.; Clark, J.; Berner, C.; McCandlish, S.; Radford, A.; Sutskever, I.; and Amodei, D. 2020. Language Models are Few-Shot Learners. In Larochelle, H.; Ranzato, M.; Hadsell, R.; Balcan, M.; and Lin, H., eds., *Advances in Neural Information Processing Systems*, volume 33, 1877–1901. Curran Associates, Inc.

Chakrabarty, T.; Padmakumar, V.; and He, H. 2022. Help me write a poem: Instruction Tuning as a Vehicle for Collaborative Poetry Writing. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*.

Chung, J. J. Y.; Kim, W.; Yoo, K. M.; Lee, H.; Adar, E.; and Chang, M. 2022. TaleBrush: sketching stories with generative pretrained language models. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*, 1–19.

Clark, E.; Ross, A. S.; Tan, C.; Ji, Y.; and Smith, N. A. 2018. Creative Writing with a Machine in the Loop: Case Studies on Slogans and Stories. In *23rd International Conference on Intelligent User Interfaces*, IUI '18, 329–340. New York, NY, USA: ACM. ISBN 978-1-4503-4945-1.

Dou, Z.-Y.; and Neubig, G. 2021. Word Alignment by Fine-tuning Embeddings on Parallel Corpora. In *Conference of the European Chapter of the Association for Computational Linguistics (EACL)*.

Fan, A.; Lewis, M.; and Dauphin, Y. 2018. Hierarchical Neural Story Generation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 889–898. Melbourne, Australia: Association for Computational Linguistics.

Fan, A.; Lewis, M.; and Dauphin, Y. 2019. Strategies for Structuring Story Generation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 2650–2660. Florence, Italy: Association for Computational Linguistics.

Flower, L.; and Hayes, J. R. 1981. A cognitive process theory of writing. *College composition and communication*, 32(4): 365–387.

Gabriel, R. P.; Chen, J.; and Nichols, J. 2015. InkWell: A Creative Writer's Creative Assistant. In *Proceedings of the 2015 ACM SIGCHI Conference on Creativity and Cognition*, 93–102. ACM.

Gero, K. I.; and Chilton, L. B. 2019. Metaphoria: An Algorithmic Companion for Metaphor Creation. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, CHI '19, 1–12. New York, NY, USA: Association for Computing Machinery. ISBN 9781450359702.

Gero, K. I.; Liu, V.; and Chilton, L. 2022. Sparks: Inspiration for Science Writing Using Language Models. In *Designing Interactive Systems Conference*, DIS '22, 1002–1019. New York, NY, USA: Association for Computing Machinery. ISBN 9781450393584.

Goldfarb-Tarrant, S.; Chakrabarty, T.; Weischedel, R.; and Peng, N. 2020. Content Planning for Neural Story Generation with Aristotelian Rescoring. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 4319–4338. Online: Association for Computational Linguistics.

Hsu, T.-Y.; Huang, C.-Y.; Hsu, Y.-C.; and Huang, T.-H. 2019. Visual Story Post-Editing. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 6581–6586.

Hua, X.; and Wang, L. 2020. PAIR: Planning and Iterative Refinement in Pre-trained Transformers for Long Text Generation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 781–793. Online: Association for Computational Linguistics.

Huang, C.-Y.; Huang, S.-H.; and Huang, T.-H. K. 2020. Heteroglossia: In-Situ Story Ideation with the Crowd. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, 1–12.

Huang, C.-Y.; and Huang, T.-H. 2021. Semantic Frame Forecast. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2702–2713. Online: Association for Computational Linguistics.

Ippolito, D.; Yuan, A.; Coenen, A.; and Burnam, S. 2022. Creative Writing with an AI-Powered Writing Assistant: Perspectives from Professional Writers. *arXiv preprint arXiv:2211.05030*.

Keskar, N. S.; McCann, B.; Varshney, L. R.; Xiong, C.; and Socher, R. 2019. Ctrl: A conditional transformer language model for controllable generation. *arXiv preprint arXiv:1909.05858*.

Kingma, D. P.; and Ba, J. 2015. Adam: A Method for Stochastic Optimization. In Bengio, Y.; and LeCun, Y., eds., *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Kryściński, W.; Rajani, N.; Agarwal, D.; Xiong, C.; and Radev, D. 2021. BookSum: A Collection of Datasets for Long-form Narrative Summarization. *arXiv preprint arXiv:2105.08209*.

Lee, M.; Liang, P.; and Yang, Q. 2022. CoAuthor: Designing a Human-AI Collaborative Writing Dataset for Exploring Language Model Capabilities. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*, CHI '22. New York, NY, USA: Association for Computing Machinery. ISBN 9781450391573.

Li, B.; Lee-Urban, S.; Johnston, G.; and Riedl, M. 2013. Story generation with crowdsourced plot graphs. In *Twenty-Seventh AAAI Conference on Artificial Intelligence*.

Li, B.; and Riedl, M. 2015. Scheherazade: Crowd-powered interactive narrative generation. In *Twenty-Ninth AAAI Conference on Artificial Intelligence*.

Lita, L. V.; Ittycheriah, A.; Roukos, S.; and Kambhatla, N. 2003. Truecasing. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, 152–159.

Loshchilov, I.; and Hutter, F. 2019. Decoupled Weight Decay Regularization. In *International Conference on Learning Representations*.

Martin, L. J.; Ammanabrolu, P.; Wang, X.; Hancock, W.; Singh, S.; Harrison, B.; and Riedl, M. O. 2018. Event representations for automated story generation with deep neural nets. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

Mirowski, P.; Mathewson, K. W.; Pittman, J.; and Evans, R. 2022. Co-writing screenplays and theatre scripts with language models: An evaluation by industry professionals. *arXiv preprint arXiv:2209.14958*.

Mori, Y.; Yamane, H.; Shimizu, R.; Mukuta, Y.; and Harada, T. 2022. COMPASS: a Creative Support System that Alerts Novelists to the Unnoticed Missing Contents. *CoRR*, abs/2202.13151.

Mostafazadeh, N.; Chambers, N.; He, X.; Parikh, D.; Batra, D.; Vanderwende, L.; Kohli, P.; and Allen, J. 2016. A Corpus and Cloze Evaluation for Deeper Understanding of Commonsense Stories. In *NAACL'16*, 839–849. San Diego, California: ACL.

Mostafazadeh, N.; Kalyanpur, A.; Moon, L.; Buchanan, D.; Berkowitz, L.; Biran, O.; and Chu-Carroll, J. 2020. GLUCOSE: GeneraLized and COntextualized Story Explanations. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 4569–4586. Online: Association for Computational Linguistics.

Padmakumar, V.; and He, H. 2022. Machine-in-the-Loop Rewriting for Creative Image Captioning. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 573–586. Seattle, United States: Association for Computational Linguistics.

Peng, N.; Ghazvininejad, M.; May, J.; and Knight, K. 2018. Towards Controllable Story Generation. In *Proceedings of the First Workshop on Storytelling*, 43–49. New Orleans, Louisiana: Association for Computational Linguistics.

Radford, A.; Wu, J.; Child, R.; Luan, D.; Amodei, D.; and Sutskever, I. 2019. Language models are unsupervised multitask learners. *OpenAI Blog*, 1(8).

Reimers, N.; and Gurevych, I. 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)*, 3982–3992. Hong Kong, China: Association for Computational Linguistics.

Riedl, M. O.; and Young, R. M. 2010. Narrative planning: Balancing plot and character. *Journal of Artificial Intelligence Research*, 39: 217–268.

Roemmele, M. 2021. Inspiration through Observation: Demonstrating the Influence of Automatically Generated Text on Creative Writing. In *12th International Conference on Computational Creativity*.

Roemmele, M.; and Gordon, A. S. 2015. Creative help: a story writing assistant. In *International Conference on Interactive Digital Storytelling*, 81–92. Springer.

Rose, S.; Engel, D.; Cramer, N.; and Cowley, W. 2010. *Automatic Keyword Extraction from Individual Documents*, 1 – 20. ISBN 9780470689646.

See, A.; Pappu, A.; Saxena, R.; Yerukola, A.; and Manning, C. D. 2019. Do Massively Pretrained Language Models Make Better Storytellers? In *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, 843–861. Hong Kong, China: Association for Computational Linguistics.

Sharma, S.; El Asri, L.; Schulz, H.; and Zumer, J. 2017. Relevance of Unsupervised Metrics in Task-Oriented Dialogue for Evaluating Natural Language Generation. *CoRR*, abs/1706.09799.

Swanson, B.; Mathewson, K.; Pietrzak, B.; Chen, S.; and Dinalescu, M. 2021. Story Centaur: Large Language Model Few Shot Learning as a Creative Writing Tool. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations*, 244–256. Online: Association for Computational Linguistics.

Xu, J.; Ren, X.; Zhang, Y.; Zeng, Q.; Cai, X.; and Sun, X. 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*, 4306–4315. Brussels, Belgium: Association for Computational Linguistics.

Yao, L.; Peng, N.; Weischedel, R.; Knight, K.; Zhao, D.; and Yan, R. 2019. Plan-and-write: Towards better automatic storytelling. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, 7378–7385.

Zhong, M.; Liu, P.; Chen, Y.; Wang, D.; Qiu, X.; and Huang, X. 2020. Extractive Summarization as Text Matching. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 6197–6208. Online: Association for Computational Linguistics.