RDF Triple-Text-Story: A Integrated Workflow for Controllable Short Story Generation

Stanford CS224N {Custom} Project

Yifu Han
Department of Energy Science and Engineering
Stanford University
yifu@stanford.edu

Yuer Zhou
Department of Civil Engineering
Stanford University
zye98@stanford.edu

Abstract

We develop an integrated RDF triple-text-story workflow to generate coherent and imaginative short stories based on given triples or prompts. It combines RDF triple-to-text close-ended generation and open-ended story generation, which allows for the incorporation of structured information about characters, settings, and events while maintaining the flexibility of free-form storytelling. To implement this approach, we fine-tune a T5 model on RDF triples-to-texts generation (using WebNLG 2020 dataset) and then a GPT-2 model on open-ended text generation (using the ROCStories and WritingPrompt datasets). We evaluate the model performance by using BLEU score and also human evaluation.

The use of two fine-tuned models, T5 and GPT-2, in an integrated workflow can help to combine the strengths of these two models. T5 is known for its ability to summarize and generate text from structured data, while GPT-2 is renowned for its ability to generate engaging and creative narratives. The results show that the triple-text-story workflow could potentially produce high-quality short stories that are both structured and engaging by leveraging the strengths of both models. Finally, we further fine-tuning GPT-2 model on the Writing Prompts dataset using various prompt engineering strategies improved the coherence, imagination, and readability of generated short stories.

1 Key Information to include

- Mentor: Siyan Li

2 Introduction

Generating coherent and imaginative short stories from given prompts is a complex task in natural language processing. While recent advancements in deep learning models and techniques have allowed for impressive progress in text generation, creating a controllable text generation model for short stories still remains a challenge. One of the primary difficulties in generating short stories is the need for a model to understand both the nuances of language and storytelling. A good short story needs to have a compelling plot, well-developed characters, and a satisfying conclusion, all of which require a deep understanding of language and narrative structure and coherence.

In this work, we present an integrated triple-to-story generation framework to produce structured and imaginative short stories based on input triples, which combines triples-to-text generation and text-to-story generation. More specifically, we propose a workflow that leverages two fine-tuned models, T5 and GPT-2, to generate coherent short stories. T5 is known for its ability to generate high-quality text from structured input (which are triples in this work), while GPT-2 is renowned for its ability to generate coherent and engaging narratives. Combining the strengths of these two models in our proposed workflow has the potential to create a highly effective and efficient model for generating short stories. By using fine-tuned T5 to generate the structured components of the story, such as character names and settings, and then using fine-tuned GPT-2 to generate the narrative itself, we can take advantage of the strengths of both models and produce stories that are both structured and engaging. The input triples are first processed by the fine-tuned T5 model, which serves as a paraphrase, generating concise sentences containing structured information about story elements. The fine-tuned T5 model (on the WebNLG dataset) benefits from its text-to-text capabilities and proficiency in various NLP tasks, such as paraphrasing and summarizing. Subsequently, the output sentences are passed to the fine-tuned GPT-2 model, which generates coherent and engaging stories using the input prompts and contextual information provided by the T5 model.

3 Related Work

See et al. [1] investigated whether the use of large, pre-trained language models such as GPT-2 and GPT-3 improves the quality of generated stories. They conducted experiments using both objective and subjective evaluation metrics, comparing the performance of pre-trained models to models trained from scratch. They found that the pre-trained models produce stories that are more coherent and better structured than the models trained from scratch, and that they are also more preferred by human evaluators. However, they also note that the pre-trained models tend to generate stories that are more stereotypical and less diverse than those generated by the models trained from scratch.

Yao et al. [2] proposed a new approach to automatic storytelling called "Plan-and-Write," which aims to improve the coherence and consistency of generated stories by explicitly modeling a story plan. The proposed approach consists of two stages: (1) planning, where the model generates a high-level story plan, and (2) writing, where the model generates the text for the story by conditioning on the plan. Fang et al. [3] proposed a new method for generating coherent and diverse stories using a hierarchical neural network model. The model generates stories in a two-step process, first generating a high-level story outline and then using it to generate the details of the story. These ideas are similar to our proposed two-stages story generation. Mao et al. [4] proposed a method to improve the quality of story generation by incorporating common sense grounding. They developed a dataset of story prompts paired with common sense statements and use it to train a language model to generate stories that are grounded in common sense knowledge. They evaluate the model using human evaluation and show that incorporating common sense grounding improves the quality and coherence of the generated stories. Similarly, in our work, we investigate whether the use of prompts improves the quality of generated stories.

4 Approach

The T5 model is fine-tuned on a well-organized and public dataset consisting of a set of resource description framework (RDF) triples (already well-extracted) representing facts about an entity or event, along with target sentences that describe the entity or event. To fine-tune T5 model on this dataset, triples are treated as input to the model, while the target sentences are treated as the output.

The GPT-2-medium model is first fine-tuned on the ROCStories dataset and then further fine-tuned on WritingPrompts datasets. The fine-tuning on the ROCStories dataset is a good way to improve the model's ability to reason about the logical structure of stories, while the WritingPrompts dataset provides a diverse range of prompts for the model to generate creative and engaging narratives. Ensembling can help improve the performance of the model and make it more robust.

The overall triples to story generation workflow is as follows:

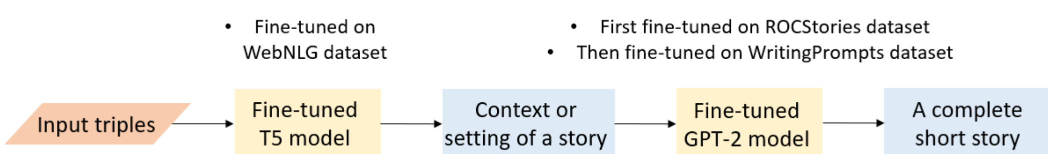

Figure 1: The proposed triples-to-story workflow.

5 Experiments

5.1 RDF Triples to Texts

5.1.1 Data

WebNLG 2020 is a dataset for Natural Language Generation (NLG) tasks, which was released in 2020 as part of the WebNLG challenge. The dataset consists of a collection of triples of structured data in RDF format, which can be converted into natural language text using various NLG techniques. The triples in the dataset are organized into 15 different categories, which cover a range of topics such as music, sports, geography, and politics. The triples are diverse and cover a wide range of relationships between entities, including properties, types, and binary relations.

5.1.2 Evaluation method

Various method can be used to evaluate the performance of generated text on training and testing dataset. We use the widely used BLEU metric given by

$$\text{BLEU} = \text{BP} \cdot \exp\left(\sum_{n=1}^{N} w_n \log p_n\right) \tag{1}$$

where BP is the brevity penalty, which is calculated as:

$$\text{BP} = \begin{cases} 1 & \text{if } c > r \\ \exp(1 - r/c) & \text{otherwise} \end{cases}$$

where c is the length of the candidate translation and r is the length of the reference translation that is closest to c in length, N is the maximum n-gram order to consider, w_n is the weight for the n-gram precision score, which is usually set to $\frac{1}{N}$, p_n is the n-gram precision score.

5.1.3 Experimental Details

As the first stage of the proposed pipeline to generate texts or prompts, we first fine-tune a T5 model on a task of RDF triple to text generation on WebNLG 2020 dataset. We reference the code (`https://github.com/MathewAlexander/T5_nlg`) to pre-process the XML data and keep the normal triples as it is and join multiple triples. We load the pre-processed data and randomly shuffle the rows to have triplets with different lengths in the training dataset. We initialize the AdamW optimizer and tune some hyper-parameters. The learning rate is set to 5×10^{-5}, and it takes 1 hours to fine-tuning the model using 35,000 training samples and 4 epochs.

5.1.4 Results

We obtain an average BLEU score of 0.64, which shows the effectiveness of the fine-tuned T5 model to reconstruct the text based on triples. In comparison, the BLEU score for the T5-base model without fine-tuning (or the baseline model) was of 0.185 (Table. 1). The fine-tuning process gives significant performance improvements. One example for the original triple in the dataset, true text, and generated text is shown below. The generated text has similar wording and meaning as the true text.

Triple: Andrews County Airport, location, Texas Texas, language, Spanish language Texas, capital, Austin Texas
Generated: Andrews County Airport is located in Texas, where Austin is the capital and Spanish is spoken.
True: Andrews County Airport is located in Texas state, whose capital city is Austin and Spanish is one of the spoken language there.

Table 1: BLEU score on WebNLG Challenge 2020 dataset.

Fine-tuned T5	Baseline (T5-base)
0.64	0.185

5.2 Generated Texts to Story

5.2.1 Data

This experiment uses a subset of 5,000 samples from the ROCStories dataset [5] for training and 200 samples for testing. The objective of the experiment is to predict the ending of a short story based on the preceding four sentences. The ROCStories dataset consists of short stories that contain five sentences each, and the task is quite challenging and open-ended.

5.2.2 Experimental Details

As the second stage of the proposed pipeline to generate story. We fine-tune a GPT-2 model for story completion by dividing each sample in the training dataset into two parts. The first four sentences are treated as the input prompt, and the last sentence is treated as the target output. The model is trained using the AdamW optimizer with a learning rate of 5×10^{-5}, a batch size of 64, and for 50 epochs. The AdamW optimizer is a variant of the Adam optimizer that incorporates weight decay regularization. The linear scheduler is used to gradually increase the learning rate from 0 to the initial value of 5×10^{-5} over a warm-up period of several epochs, after which the learning rate was kept constant at the initial value for the remaining epochs. This technique is commonly used to improve the stability and convergence of deep learning models during training. In addition, the accumulating batch size approach was used to manage the large size of the GPT-2 model by dividing the training data into smaller batches and accumulating gradients across several batches before performing a parameter update. This approach can help to reduce the memory requirements of training deep learning models with large numbers of parameters.

We also compare the performance of different fine-tuning techniques. Specifically, this experiment tests the performance of fine-tuning the entire model, the last 6 layers, or the last 2 layers of GPT-2-small. In addition, we compares the performance of different GPT-2 model architectures, including GPT-2-small with 12 layers, GPT-2-medium with 24 layer, and GPT-2-large with 36 layer, to gain insights into the trade-offs between model complexity and performance on our task.

5.2.3 Results

The performance of the model is evaluated by generating endings for the test samples, which are then assessed for quality using both BLEU score and human evaluation. The results of the experiment suggest that fine-tuning the GPT-2 model leads to a significant increase in BLEU score compared to the baseline pre-trained GPT-2-medium model. This suggests that the fine-tuning step is effective in improving the quality of the generated text. Interestingly, the results in Table 2 indicate that the fine-tuning techniques, including fine-tuning the entire model, the last 6 layers, or the last 2 layers of GPT-2-small, have relatively small impact on the BLEU score on the testing dataset. In addition, the results in Table 3 suggest that increasing model complexity can lead to a slight improvement in the BLEU score on the testing dataset. However, it is important to note that increasing model complexity comes at a cost, both in terms of computational resources required for training (e.g., it only takes 1 hour to fine-tune a GPT-2-medium model on a NVIDIA A100 GPU in our task) and in terms of model size. We have decided to use the GPT-2-medium model as it strikes a good balance between model complexity and performance on the testing dataset, as measured by the BLEU score.

While the step of the generated texts or prompts to a story using GPT-2-medium has a relatively low BLEU score of 0.15 (Table 3), it is important to note that this task is more open-ended compared to the previous step of converting RDF triples to texts or prompts, which had a higher BLEU score of 0.64. Most importantly, the generated endings are rated as coherent and plausible by human evaluators in most cases, as illustrated in Table 4. These results suggest that while our model may not generate the exact same true endings, it is still capable of producing contextually relevant and coherent text that conveys same sentiment. We can generate following story based on given context.

Table 2: BLUE score on ROCStories dataset.

Model	Description	BULE score
GPT-2-medium	**Baseline without fine-tuning**	0.029
GPT-2-small	Fine-tune **entire model**	0.124
GPT-2-small	Fine-tune **last 6 layers**	0.114
GPT-2-small	Fine-tune **last 2 layers**	0.106

Table 3: BULE score on ROCStories dataset.

Model	Description	BULE score
GPT-2-medium	**Baseline without fine-tuning**	0.029
GPT-2-medium	Fine-tune entire model, batch size of 16	0.115
GPT-2-medium	Fine-tune entire model, batch size of 64	0.155
GPT-2-medium	Fine-tune entire model, 100 epoch, batch size of 128	0.156
GPT-2-large	Fine-tune entire model, batch size of 64	**0.183**

5.3 Further Investigations on Fine-tuning GPT-2 Model (based on model in 5.2)

5.3.1 Data

We use the Writing Prompt dataset (Fan et al., 2018)[3] to further fine-tune our story generation model (based on already fine-tuned model in 5.2), focusing on hierarchical story generation. The dataset is composed of over 300,000 human-written stories, each accompanied by a prompt (sourced from an online forum). This dataset is diverse in terms of topics, lengths, and details, with prompts inspiring multiple story responses. We use part of the overall dataset, and partition it into 80,000 training samples, 2,000 validation samples, and 1,000 test samples for story model fine-tuning. In order to improve model performance, we pre-process the dataset using the Hugging Face's Transformers library's summarizing pipeline. This pipeline helps us to condense the original stories while retaining their core ideas and essential details, which makes them more suitable for training our model.

5.3.2 Experimental details

Prompt Engineering

To further improve the coherence and readability of the generated short stories (before the actual fine-tuning), we first conduct experiments using various prompt engineering strategies and assess their impact on model performance with identical configurations. The prompt engineering strategy (selection) experiments involve a relatively small dataset consisting of 2,000 training, 300 validation, and 300 test samples (not the entire training dataset due to computational resources limitation). The strategies we investigate include: (1) providing instructions and separators to guide the model's focus; (2) summarizing input prompts to emphasize key aspects; (3) incorporating sub-prompts to direct model structure and style; (4) implementing an iterative refinement technique that generates the story in smaller chunks, then using the output of each chunk as input for the next chunk; and (5) extracting the most important keywords from the prompt using nltk library and incorporating them into input.

This experiment employs the already fine-tuned GPT-2-medium model (in section 5.2) with a learning rate of 5×10^{-5}, 100 warm-up steps, a batch size of 16, and 5 epochs, and we further fine-tune it on the Writing Prompt dataset. The training time varies for different prompt engineering strategies, with the iterative refinement strategy taking longer due to its iterative nature (approximately 2 hours). The other strategies take similar training time (approximately half an hour). The training and validation losses of different prompt strategies are shown in Fig. 2 Additionally, we also compare their performance using human evaluation and BLUE scores, as shown in Tables. 9 and Table. 10.

The experimental results demonstrate that the "Context and Clarity Sub-prompts" strategy generate the most coherent, imaginative, and readable stories in relation to the given prompt, achieving the highest BLUE score and the most stable validation loss. The "Instructions and Separators" method generated more imaginative stories than the base model (the fine-tuned GPT2-medium model in section 5.2 without any prompt engineering) and achieved the second-highest BLUE score. The base model generated stories that are more straightforward and directly connected to the prompt without

Table 4: Examples of story generation on ROCStories dataset.

Given texts	True endings	Predicted endings
A boy was looking through the window of a pet store. He couldn’t decide between a cat or dog. Suddenly, a dog licked the window. The cat lounged around and was lazy.	The boy picked the dog in the end.	The boy decided to pick a dog and bought it.
Lorraine hasn’t seen her brother in two Year’s. She decides it is finally time for a visit. Lorraine takes the long drive to Idaho where her brother lives.	When the visit is over, Lorraine feels happy to be home.	Lorraine feels better that she can spend time with her brother.

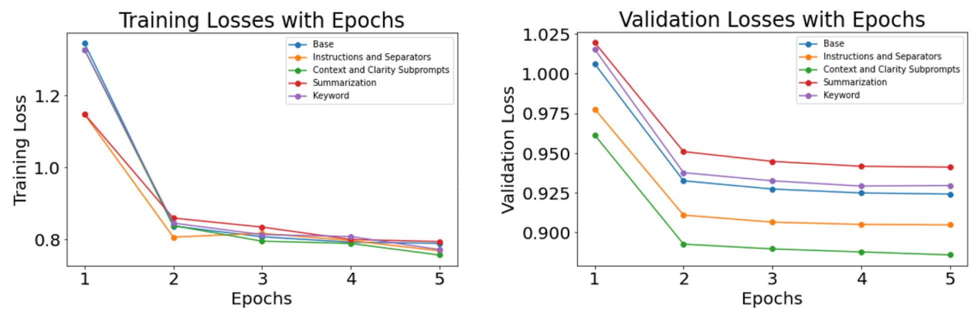

Figure 2: Training and Validation losses of different prompt strategies on WritingPrompts dataset.

emotion and sense of adventure present in some of the other stories. However, the "Summarization", "Iterative Refinement" and "Keyword Extraction" methods resulted in poor performance, with the generated stories being less coherent and having no clear connection to the given prompt.

Overall, our experiments show that prompt engineering strategies, especially the Sub-prompts and Instructions and separator methods, can improve the coherence, imagination, and readability of the generated short stories. It is worth noting that due to the length of the generated stories, we only provide the performance of the best prompt engineering method in Table 5, and a complete comparison of the performance of different prompt engineering strategies is provided in the Appendix.

Training Process

Table 5: Performance of prompt engineering (Context and Clarity Sub-prompts).

Context and Clarity Sub-prompts	A magical book transports a child to a different world. [Write a story that is engaging, imaginative, and easy to follow. Use clear and concise language.][Focus on vivid descriptions, well-developed characters, and a strong plot.]	For the past two years, every night she had prayed. The day she took this book with her out of the box she ’d cried. She would have cried again if she didn’t read, but she hadn’t. It wasn’t just the book, it was the promise. There were no more excuses. She was reading the next page, the next word, when the boy in the book asked her a question. Her eyes widened. The boy was five-years old. “Can we go back home?” I want you to read the book,” she whispered, almost inaudibly, I want you to know what’s really happening, I want you to know that everything in this world is fake.” I don’t understand.” he said, How can you possibly know? I want to go home!”	0.175

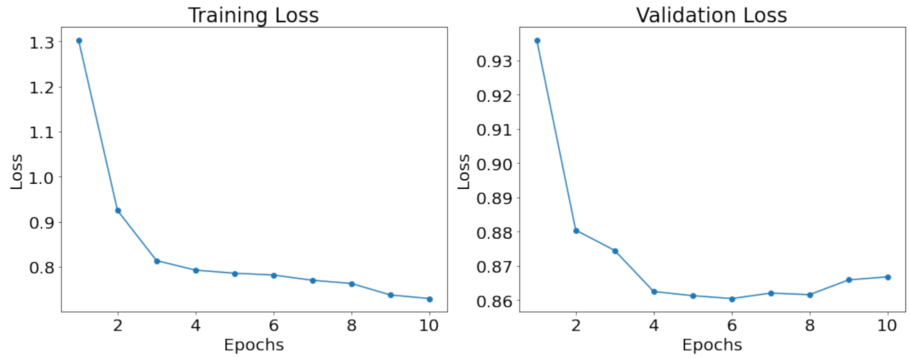

Figure 3: Training and validation losses with epochs on WritingPrompts dataset.

In our training experiment, we fine-tune the final model with above chosen prompt strategy using various hyper-parameters to find the optimal combination. We test multiple combinations of hyper-parameters, including batch sizes (16, 32, 64, 128), learning rates (5×10^{-5}, 1×10^{-4}, 5×10^{-4}, 1×10^{-3}), and warm-up steps (200, 500, and 1000).

During the training process, we observe that the model is prone to overfitting. While the training losses decrease continuously, the validation loss increase easily. To tackle this issue, we monitor the validation loss and stop training if the validation loss does not improve for a specified number of consecutive epochs (patience parameter is set to 3 in our case). To prevent exploding gradients, we clip the gradients of the model parameters using a maximum gradient norm of 1.0. We also introduce weight decay of 0.01 in the AdamW optimizer to add an L_2 regularization term to the model parameters. Moreover, We use a linear learning rate scheduler with warm-up steps to gradually increase the learning rate during the initial phase of training and then decrease it over time. Additionally, we concatenate the tensors while considering the maximum sequence length constraint to accommodate the large size of the GPT-2 model, allowing us to effectively process the input data. After iterating through various hyper-parameter combinations, we select the best model based on the lowest validation loss, the final model configuration use a batch size of 64, 10 epochs, a learning rate of 1×10^{-4}, 200 warm-up steps, 4 gradient accumulation steps, a weight decay of 0.02, an Adam epsilon of 1×10^{-8}, and a max gradient norm of 1.0. The training takes approximately 2 hours.

5.3.3 Results

The quantitative results obtained from the further fine-tuned model are presented in the form of training losses, validation losses, and BLUE score. A comparison with the base model indicates improvements in story generation quality, and the results are shown in Fig. 3 and Table. 6

Table 6: BLEU scores on WritingPrompts dataset.

Further fine-tuned GPT-2 model	Fine-tuned GPT-2 model in section 5.2
0.188	0.127

6 Analysis

6.1 Two-stage: RDF Triples to Story

Once the two separate models have been fine-tuned, then they can be incorporated into a two-stage story generation workflow. The performance of the developed workflow (for the case of GPT-2-medium model fine-tuned on ROCStories dataset) is illustrated through three random examples of input RDF triples, generated text, and the resulting story, which are shown in Table 7. The generated stories are contextually relevant and coherent to the texts and original triples. More importantly, we can achieve even better story generation performance after we further fine-tune the trained model (in section 5.2) using Writing Prompt dataset. The generated example story is shown in Table 8, which is a high-quality short story that is both structured and engaging. The coherence, imagination, and readability of generated short stories are further improved.

Table 7: Example 1, 2 and 3: RDF triples to text and then a story in two-stages (T5 model and GPT-2-medium model in section 5.2)

<table>
<tr><td>Example 1 triples: Vermont | largestCity | Burlington,_Vermont
Alvah_Sabin | state | Vermont.</td></tr>
<tr><td>Text: Alvah Sabin was from Vermont, whose largest city is Burlington.</td></tr>
<tr><td>Story: Alvah Sabin was from Vermont, whose largest city is Burlington. After many years, Vermont became very expensive. Avah could no longer afford to live there. He and his wife moved to Florida. Avah has lived there for 20 years now.</td></tr>
<tr><td>Example 2 triples: Buzz_Aldrin | birthPlace | Glen_Ridge,_New_Jersey
Buzz_Aldrin | mission | Apollo_11
Buzz_Aldrin | status | "Retired"
Apollo_11 | operator | NASA</td></tr>
<tr><td>Text: Buzz Aldrin, who was born in Glen Ridge, New Jersey, was part of Apollo 11 as chosen by NASA, and is now retired.</td></tr>
<tr><td>Story: Buzz Aldrin, who was born in Glen Ridge, New Jersey, was part of Apollo 11 as chosen by NASA, and is now retired. He has visited the moon with Neil Armstrong. Aldrin still loves to go to the moon, and wants to see it restored. He has spoken with President Bush about returning to the moon. Aldrin recently wrote a book about his adventures on the moon.</td></tr>
<tr><td>Example 3 triples: Ace_Wilder | background | "solo_singer"
Ace_Wilder | occupation | Songwriter
Ace_Wilder | birthPlace | Sweden</td></tr>
<tr><td>Text: Sweden is the birthplace of Ace Wilder, who has a background as a solo singer.</td></tr>
<tr><td>Story: Sweden is the birthplace of Ace Wilder, who has a background as a solo singer. Wilder's first big break came when he was fourteen. He had a hit in the country song contest. He became very rich performing there.</td></tr>
</table>

Table 8: Example: RDF triples to text and then a story in two-stages (T5 model and further fine-tuned GPT-2-medium model in section 5.3)

<table>
<tr><td>Example triples: Ace_Wilder | background | "solo_singer"
Ace_Wilder | occupation | Songwriter
Ace_Wilder | birthPlace | Sweden</td></tr>
<tr><td>Text: Sweden is the birthplace of Ace Wilder, who has a background as a solo singer.</td></tr>
<tr><td>Story: Sweden is the birthplace of Ace Wilder, who has a background as a solo singer. One day, he went into the recording studios and found the most talented voice he had ever heard. He then hired the best guitarists he could find. After three years, he was able to record and release his first solo album "The Man of a Thousand Pounds". He then sold his second album and is now in the process of doing a live album entitled "Ace's Next Concert" with his orchestra. He now has a full orchestra that plays every song on record.</td></tr>
</table>

7 Conclusion

This project focuses on generating short stories from RDF triples integrating both T5 and GPT-2 language model. Fine-tuning the GPT-2 model leads to significant improvements in the quality of generated text, as measured by BLEU score and human evaluation. Further fine-tuning on the Writing Prompts dataset using various prompt engineering strategies improved the coherence, imagination, and readability of generated short stories. Our two-stage workflow of RDF triples to text and then a story, using fine-tuned T5 model and GPT-2 models, produces contextually relevant and coherent stories. The primary limitation is the quality of input RDF triples and prompts, which may affect the quality of the generated stories. Future work could focus on improving the quality of input data and exploring other language models to generate longer and more complex stories.

References

[1] Abigail See, Aneesh Pappu, Rohun Saxena, Akhila Yerukola, and Christopher D Manning. Do massively pretrained language models make better storytellers? *arXiv preprint arXiv:1909.10705*, 2019.

[2] Lili Yao, Nanyun Peng, Ralph Weischedel, Kevin Knight, Dongyan Zhao, and Rui Yan. Plan-and-write: Towards better automatic storytelling. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 7378–7385, 2019.

[3] Angela Fan, Mike Lewis, and Yann Dauphin. Hierarchical neural story generation. *arXiv preprint arXiv:1805.04833*, 2018.

[4] Huanru Henry Mao, Bodhisattwa Prasad Majumder, Julian McAuley, and Garrison W Cottrell. Improving neural story generation by targeted common sense grounding. *arXiv preprint arXiv:1908.09451*, 2019.

[5] Nasrin Mostafazadeh, Nathanael Chambers, Xiaodong He, Devi Parikh, Dhruv Batra, Lucy Vanderwende, Pushmeet Kohli, and James Allen. 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, 2016.

A Appendix (optional)

Table 9: Performance of prompt engineering strategies.

Strategies	Input prompt	Generated stories	BLUE scores
Base	A magical book transports a child to a different world.	A child stood in a dimly lit room, curtains drawn, with a book in front of him. He examined the drawings, glancing around the room until his gaze settled on the book's dark wooden cover. He reached out and touched it. A voice emerged from behind the cover. "Good morning, child." The child looked at the book. "What kind of book is this?" "This is a magical book," the voice replied. The child's eyes widened. "Really?" "Yes, this magical book contains the story of your world."	0.129
Instructions and Separators	[Please generate a story that revolves around:] A magical book transports a child to a different world. [separator] [Here is the generated story:] ... [END]	All the children's eyes were glued to the book, awestruck. The title seemed to speak to them, as if revealing a secret. "There are worlds," said one child, picking up the book, his face pressed against it. "There are other worlds." The book flipped open, words appearing as if carved into the pages. "Wherever you go, there you go," it read. The boy gazed at the sky, as if looking into a mirror. "Wherever you go, there you go," he repeated. "No more. You will know," the book whispered, now glowing on its surface. One by one, the other children stood up and left. "Ahh, here we go," said the boy.	0.157

Table 10: Performance of prompt engineering strategies.

Subprompts	A magical book transports a child to a different world. [Write a story that is engaging, imaginative, and easy to follow. Use clear and concise language.][Focus on vivid descriptions, well-developed characters, and a strong plot.]	For the past two years, every night she had prayed. The day she took this book with her out of the box she 'd cried. She would have cried again if she didn't read, but she hadn't. It wasn't just the book, it was the promise. There were no more excuses. She was reading the next page, the next word, when the boy in the book asked her a question. Her eyes widened. The boy was five-years old. "Can we go back home?" I want you to read the book," she whispered, almost inaudibly, I want you to know what's really happening, I want you to know that everything in this world is fake." I don't understand." he said, How can you possibly know? I want to go home!"	0.172
Keyword Extraction	[magical book transports child different] A magical book transports a child to a different world.	"Hey Mike!" A voice came from behind me, startling me for a second. My head snapped up at the sound of it, startling me even more so. "Hello, how's it going?" I asked, trying to keep calm. It's always the same... "Okay... so you said you couldn't handle this anymore." "I am no longer able to control my powers anymore." I quickly turned my head around to see what the commotion was about. The children in the room were sitting down to talk, and the one in the front of the room had his phone out and was playing some video game.	0.125