Consistency-Enhanced Story Generation

Team Members: Debashish Roy (2021201034)
Professor: Dr. Manish Shrivastava
Mentor: Mr. Bhavyajeet Singh
17th November 2022

Introduction:

We have used pretrained architectures GPT-2 for story generation. GPT-2 often struggles to maintain consistency and often ends up generating content incoherent with the given prompt for a generation. Story generation revolves around some common character and revolves around a plot. It is a more difficult task than text generation due to the following reasons:

- It must maintain consistent plots to create a plausible tale.
- It must ensure that the characters remain constant throughout the story, and
- It should be concerned with the coherence of the text units, such as phrases or sentences.

We are using a 2-step approach to story generation involving the generation of a plot outline based on the prompt followed by the actual generation of the story.

Existing Approaches:

- In recent years, a number of end-to-end methods based on Sequence-to-Sequence models have been presented. These methods may instantly construct a left-to-right tale. These data-driven techniques can produce tales in natural language without the need for other abstract representations. The high-level relationships between the plot elements and the consistency of the plots across the novel are difficult to depict with these tools, though.
- These models often break down the process of creating stories into two stages: creating the middle form first, and then creating the finished tale. These approaches employ a variety of middle conditions, including keywords, phrases, and event tuples.
- **Related Work:** I have worked with the paper mentioned in the project description and worked on implementing the paper Consistency and Coherency Enhanced Story Generation, ECIR 2021.

Dataset:

We are using the Writing Prompt Dataset by FAIR. This is used to generate consistent stories by Hierarchical Neural Story Generation (Fan et al., 2018) https://arxiv.org/abs/1805.04833. Data is separated into prompt and story. This is extracted from **Reddit,** where one user will come up with a prompt and a single story and another will write a story revolving around that prompt (https://www.reddit.com/r/WritingPrompts/).

The dataset is already divided into test, train, and validate. Test.wp_source contains prompts and corresponding to that test.wp_target contains stories. The size of the complete data is around 1 GB. The data is hosted on https://www.kaggle.com/datasets/ratthachat/writing-prompts.

Converting Data to suitable format:

We have converted the dataset in the format The format of the dataset is
[WP] PROMPT <endprompt> OUTLINE <endoutline> TOP 10 KEYWORD/ PHRASE.
Where the outline is generated using the Text Rank algorithm. And Top 10 keywords are found using the RAKE algorithm.

Architecture:

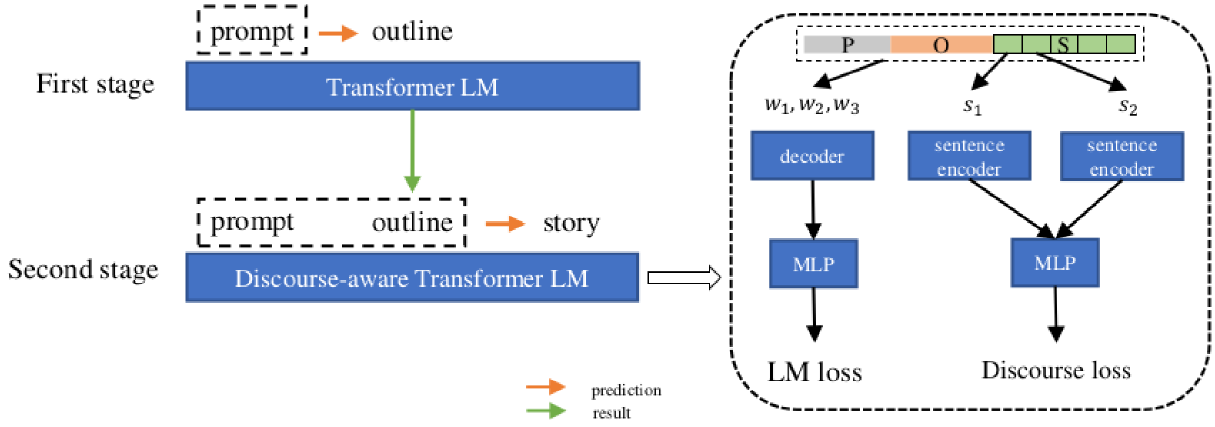


Figure 1: The framework of our model for story generation.

The approach is mainly divided into two parts with one enhancement:

- **Prompt to Outline Generation:**
    - A Transformer-based language model-based decoder is used to generate outlines. The outline is generated by taking a variation of the Test Rank Algorithm to extract the abstract of the story and then taking 30% of the abstract.
    - Then we concatenate prompt X and outline Z with <SEP> token to get a sequence X0.
    - For training, we compute the cross-entropy of all tokens in X0 as a normal language model. When testing, given the prompt tokens as context, the decoder generates outline tokens.

- **Prompt and Outline to Story Generation:**
    - Another decoder with the same architecture is used to generate stories.
    - We concatenate prompt X, outline Z, and story Y with <S> and <SEP> tokens to get a sequence X1.
    - For training, we compute the cross-entropy of prompt and story tokens in X1.

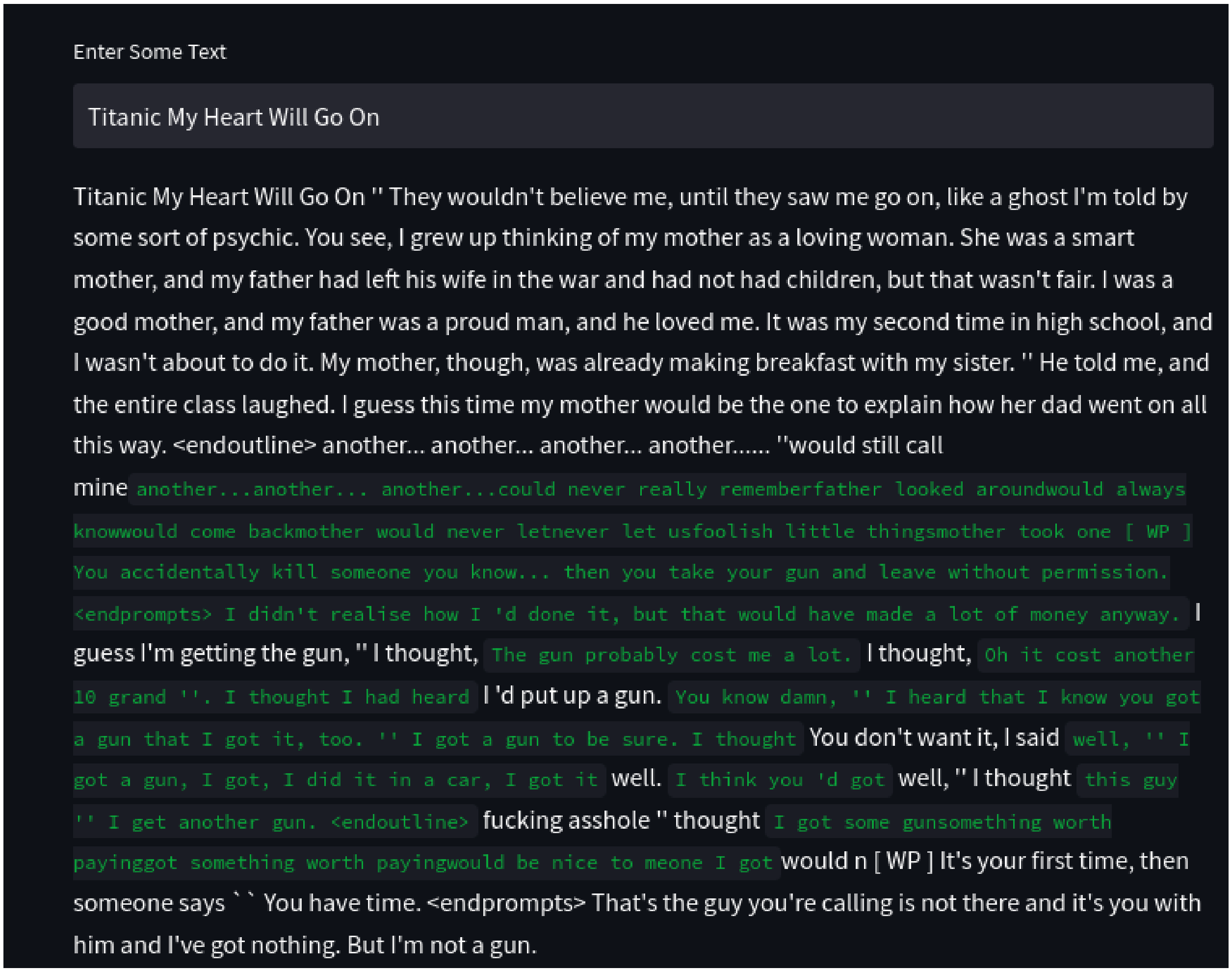
- **Training:** We have used the gpt2 model and finetuned it as per our use case. It is used to generate outlines. We concatenate prompt X and outline Z with <SEP> token to get a sequence X0. For training, we compute the cross-entropy of all tokens in X0 as a normal language model.
- **Testing:** While testing has given the prompt tokens as context, the model generates outline tokens.

Top-k and Top-p

- I have used **Top-k and Top-p** selection strategies. Top-k allows us to choose the next word from top-k probability, and then sample from it. Whereas Top-p picks amongst the top tokens whose probabilities add up to p.

Deployment

Link to the Code (https://github.com/debashish05/Story-Generation-Using-GPT2) to run inference using streamlit and complete code.



Enter Some Text

Titanic My Heart Will Go On

Titanic My Heart Will Go On '' They wouldn't believe me, until they saw me go on, like a ghost I'm told by some sort of psychic. You see, I grew up thinking of my mother as a loving woman. She was a smart mother, and my father had left his wife in the war and had not had children, but that wasn't fair. I was a good mother, and my father was a proud man, and he loved me. It was my second time in high school, and I wasn't about to do it. My mother, though, was already making breakfast with my sister. '' He told me, and the entire class laughed. I guess this time my mother would be the one to explain how her dad went on all this way. <endoutline> another... another... another... another...... ''would still call mine another...another... another...could never really rememberfather looked aroundwould always knowwould come backmother would never letnever let usfoolish little thingsmother took one [WP] You accidentally kill someone you know... then you take your gun and leave without permission. <endprompts> I didn't realise how I 'd done it, but that would have made a lot of money anyway. I guess I'm getting the gun, '' I thought, The gun probably cost me a lot. I thought, Oh it cost another 10 grand ''. I thought I had heard I 'd put up a gun. You know damn, '' I heard that I know you got a gun that I got it, too. '' I got a gun to be sure. I thought You don't want it, I said well, '' I got a gun, I got, I did it in a car, I got it well. I think you 'd got well, '' I thought this guy '' I get another gun. <endoutline> fucking asshole '' thought I got some gunsomething worth payinggot something worth payingwould be nice to meone I got would n [WP] It's your first time, then someone says ` ` You have time. <endprompts> That's the guy you're calling is not there and it's you with him and I've got nothing. But I'm not a gun.

Training and Evaluation Loss:

Step	Training Loss	Validation Loss
10000	3.716700	3.637604
20000	3.668300	3.597819
30000	3.648300	3.576489
40000	3.618600	3.563454

No. of sample trained on is 732746.

Evaluation Mechanism and Results:

- **ROUGE Recall:** is the ratio of no. of overlapping words to the no of words in the reference summary.
- **ROUGE Precision:** It is the ratio of no. of overlapping words to the length of the summary.
- **ROUGE-2:** It measures bigram overlap between two sentences.
- **ROUGE-L:** It measures the longest matching sequence of words using LCS.
- **ROUGE-F1:** It is the harmonic mean of precision and recall.

S.No.	Metric	Value
1.	Rouge-L-P	0.27997
2.	Rouge-L-R	100.0
3.	Rouge-L-F	100.0
4.	Rouge-2-P	0.00113
5.	Rouge-2-R	0.22955
6.	Rouge-2-F	0.22955

Analysis

- Mentor suggested analyzing validation loss along with the training loss, to get an idea that the model is learning anything or not. This is a standard procedure to test whether data is overfitting or not. If training loss kept on decreasing while validation loss kept constant, that means the model is overfitted.
- Since both are decreasing we reached the conclusion that it is learning something.

- The result given by the model has a high Rouge-L-R score since the model at least replicates what the prompt is provided and considering it in the longest common subsequence we get a perfect match.
- Accuracy needs to be improved. One of the causes is that the coherence between the sentences is not too high.
- Rouge score is not perfect. A story can be written in many forms with the same meaning and it will give a high score with the words that it will match.
- More training is required. One epoch is taking 4 hours of training and 1 hour of validation. And the number of parameters is high. So if we can train on multiple epochs the result would have definitely increased.

Reference:

- Consistency and Coherency Enhanced Story Generation, ECIR 2021 https://arxiv.org/abs/2010.08822
- https://www.kaggle.com/datasets/ratthachat/writing-prompts
- https://arxiv.org/abs/1805.04833
- https://www.reddit.com/r/WritingPrompts
- Text Rank (Summarization): https://radimrehurek.com/gensim_3.8.3/summarization/summariser.html
- Rake Algorithm (Keyword extractor): https://pypi.org/project/rake-nltk/
- Finetuning GPT Link Link