Generating Short Science Fiction with Language Model

Bin Hu

May 3, 2023

1 Introduction

Generative language models have been an important topic in natural language processing. Current large language models (LLM), such as GPT-3, can perform well in zero-shot learning, partially due to their large number of parameters and training on massive corpora[1]. However, zero-shot learning and even few-shot learning are challenging for smaller models that can be trained on personal computers.

In this project, we want to study how well such small language models which can be trained on PC can imitate and generate text with a specific style. In our experiment, we try to train a customized GPT-2 model on the corpus of the SCP-Foundation series and use a unified prompt to generate short science fiction stories in a similar style. We will evaluate the generated tasks both by human and algorithms metrics. Python is the main programming language in this project and almost 7000 SCP-Foundation articles are used as the corpus to build the model.

2 Method

2.1 Corpus

We use the dataset called *SCP 001 to 6999*[1] on Kaggle which is collected by scraping through the SCP-Wiki[2]. There are 6999 SCP-Foundation series articles collected in the dataset (SCP-001 to SCP-6999) and each article has 7 attributes in the dataset. We will only use the "title" and "content" which add up to the main body of the article.

The SCP-Foundation series is a collaborative writing project of science fiction. All articles share a universe where an organization (SCP-Foundation) makes efforts to contain and study various anomalies (or SCPs). These articles are typically written in a scientific or documentary style and often involve horror, suspense, or supernatural elements. The distinctive

[1] https://kaggle.com/datasets/czzzzzzz/scp1to7
[2] https://scp-wiki.wikidot.com

and consistent worldview and article structure of this series makes it an excellent object for text imitation.

A typical SCP article consists of specific sections: title, code, object class, special containment procedures, and description. Besides, some tricks like data removal marks are included to create a sense of mystery.

2.2 Language model

We use the pre-trained *GPT2-medium* model from huggingface[3], which is the 355 million parameter version of the original GPT-2 (the original version has 1.5 billion parameters).

GPT-2 is a transformer decoder-only neural network that serves as a language model, and the self-attention mechanism makes it able to model long-term dependencies in the context, thus being able to generate long coherent text compared than the traditional n-gram models[3]. GPT-2 model is pre-trained on a really large corpus of English data scraped from web pages in a self-supervised fashion.

2.3 Evaluation

The evaluation consists of two parts: Human evaluation and algorithm evaluation.

Human evaluation will focus on if the generated texts are generally similar to the original texts in paragraphs, tones, and possible drawbacks.

Algorithm evaluation will focus on two scores, the Flesch Kincaid readability score and the original pre-trained GPT-2 model's perplexity score on both generated text and valid set. The Flesch Kincaid Grade Level is a readability metric that measures the approximate grade level needed to understand a piece of text[2]. It uses a formula that takes into account the average sentence length and the average number of syllables per word in a text.

We will generate 100 articles with each designed prompt for evaluation which will be explained in the next section. If the scores on the generated texts are close to the scores on the valid set, we can suppose the model learns the style well and the generated articles are difficult to be recognized.

3 Experiment

3.1 Data pre-processing

The data pre-processing consists of two parts, data cleaning, and format standardization. The data pre-processing makes sure that the model will be trained on a corpus in a unified format,

[3]https://huggingface.co/gpt2-medium

which is important for the model to learn the structure and style of the corpus. And we can easily choose the prompt in the next steps.

Data cleaning will first be applied to the content of the article to remove the web content that is not related to the article. Then the title and the content will be formatted as follows:

Listing 1: Format standardization

```
This is an SCP-Foundation friction:
Title: ***
Item #: ***
Special Containment Procedures: ***
Description: ***
```

3.2 Model Training

Model	GPT-2 Medium
Tokenizer	GPT-2 Medium tokenizer
Padding token	End-of-sequence (EOS) token
Padding length	256 tokens
Batch size	8
Optimizer	AdamW with learning rate of 1e-4
Scheduler	StepLR with step size of 1 and gamma of 0.1
Epochs	2

Table 1: Model configuration details

We use Python as the programming language and PyTorch as the deep learning framework to train the model. The code runs on a Kaggle machine with a Tesla P100 GPU.

The corpus will be randomly divided into a training set with 6000 articles and a valid set with 999 articles. We will calculate the model's perplexity score on the valid set after every training epoch, and the model with the lowest perplexity score will be selected for text generation. The perplexity score describes the degree of uncertainty of the language model in predicting the next word in a sequence. It is calculated by taking the exponential of the cross-entropy loss of the model on the validation set, hence lower is better.

The configuration of the training is listed in Table 1.

Training epoch	**Perplexity on valid set**
0 (pre-trained)	22.891
1	**8.773**
2	8.859

Table 2: Perplexity before training / after every training epochs

The perplexity score after every training epoch is listed in Table 2 above, and the perplexity score after epoch 1 is the lowest, hence the model after the first training epoch will be saved for generation in the next section.

3.3 Generation

The language model works by predicting the possibility distribution to generate the next word (token) given the content in the front. Hence prompts are important to text generation. We selected two prompts as a part of the unified format of the training set:

Listing 2: Prompt without title

```
This is an SCP-Foundation friction:
Title:
```

Listing 3: Prompt with title

```
This is an SCP-Foundation friction:
Title: The angry linguist
```

The difference between the two prompts is that the first prompt asks the model to do the open title generation. The model should come up with the title itself. And the second prompt has already provided the model with a title. The model is supposed to write an article that is related to the given title.

We use top-p sampling to select the token when generating the text, which chooses from the smallest possible set of words whose cumulative probability exceeds probability p. We set the p as 0.9. Compared to the search methods, top-p sampling generates more diverse texts.

Because we did not add the "stop token" when training, the model can theoretically produce an unlimited length of text. Hence we set the maximum token of the generated text to 384. 100 articles are generated for each prompt. All generated texts are available at the GitHub repository and the link will be included in the appendix.

We also generated 100 articles with the GPT-2 model that is not trained on our corpus (only pre-trained) using the prompt with the title as a comparison for algorithm evaluation.

4 Evaluation

4.1 Human Evaluation

Most of the generated articles fit nicely into the structure of a typical SCP-Foundation article: They consist of a title, code, object class, special containment procedures, and description.

And some data deletion markers appear in the article at the proper places which is expected. However, many issues can be noticed.

The first issue is digression. Some articles generated given the title ”The angry linguist” appears not related to the title in content. This issue is also found in some articles open-generated (even though the model makes the title itself). A possible explanation is that the GPT-2 model we use is relatively small and can not model the long-term dependency well. It could become better if we switch the position of the ”special containment measure” and ”description” in the unified format so that the important section ”description” can get closer to the title.

Another issue is self-inconsistency. Here is an example: “SCP-6633 is a 5m x 5m humanoid containment chamber... It is to be locked with a keycard to the chamber...” Also, the class of the object sometimes contradicts the description. We suppose this is because as a relatively small language model, our GPT-2 can hardly understand the commonsense in the real world. A larger corpus to pre-train on may help the model to improve.

Besides, the repeated pattern is an issue that appears in some generated articles. Some identical sentences appear multiple times in the article and make no sense. We use the sampling method to alleviate this issue but it still appears sometimes. Multiple attempts to generate and delete articles where repeating patterns occur can be a possible solution.

4.2 Algorithm Evaluation

The average Flesch Kincaid readability score of the valid set, 100 articles generated by the model trained on our corpus with each prompt, and 100 articles generated by the model that is only pre-trained with a prompt with the title are listed in Table 3.

Articles	**Average Flesch Kincaid readability score**
Valid set	9.72
100 generated article, w/o title	10.46
100 generation, w/ title	10.38
100 generation, only pre-trained, w/ title	23.69

Table 3: Flesch Kincaid Readability score of the articles

Obviously, the model trained on the corpus can produce the article with closer readability to the articles in the valid set than the model that is not trained on our corpus, which indicates the generated articles by the trained model are similar in vocabulary and the length of sentences to the real SCP articles written by human. The model does learn how to write readable text in training.

The average perplexity score evaluated by the GPT-2 model that is only pre-trained on the valid set, 100 articles generated by the model trained on our corpus with each prompt,

and 100 articles generated by the model that is only pre-trained with a prompt with the title are listed in Table 4.

Articles	**Average Perplexity**
Valid set	22.89
100 generated article, w/o title	8.86
100 generation, w/ title	9.00
100 generation, only pre-trained, w/ title	9.17

Table 4: Flesch Kincaid Readability score of the articles

It is plain to see that the perplexity score of the text generated by the trained model is much lower than the perplexity score of the valid set articles, and is close to the perplexity score of the model which is not trained on our corpus. It indicates that the generated texts can be easily distinguished by the pre-trained GPT-2 model even though the model we use to generate the text is trained on a corpus in a specific style, which is because of the language model's essence as a possibility distribution.

4.3 Further research

Future work can focus on comparing different sizes of networks on different sizes of training corpus and adding the stop token for text generation. More evaluation metrics can be used in algorithm evaluation such as fluency and coherence.

5 Team self-assessment

Bin Hu is the sole member of this team. He completed the code and the report, and all the rest work.

References

[1] BROWN, T. B., MANN, B., RYDER, N., SUBBIAH, M., KAPLAN, J., 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. M., 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. Language models are few-shot learners, 2020.

[2] FLESCH, R. Flesch-kincaid readability test. *Retrieved October 26*, 3 (2007), 2007.

[3] JUFARSKY, D., MARTIN, J. H., JUFARSKY, D., AND MARTIN, J. H. Speech and language processing: an introduction to natural language processing, computational linguistics, and speech recognition. *Pearson education, Asia* (2000).

A Appendix

A.1 Code and full generated examples

The code and generated texts in this project are available in GitHub repository. Please refer to the README.md before running the code.

https://github.com/binhu02/LING5801_Project

A.2 Selected generated samples 1

Note that the bold text is the prompt. The last truncated sentence is removed manually.

This is an SCP-Foundation fiction:

Title: The angry linguist

Item #: SCP-2352

Object Class: Safe

Special Containment Procedures: The property containing SCP-2352 has been acquired by the Foundation and designated Site-2352. A perimeter has been established around the property and the surrounding area is to be blocked off from civilian use and monitored by a security team. Any suspicious activity or property damage is to be reported to the current Level-1 researcher. No further containment procedures are required at this time.

Description: SCP-2352 is a small, dilapidated farmhouse located in the small village of [redacted], Missouri. On the right side of the property is a large, dilapidated barn. This has been boarded up, however a door with no door leading outside has been left open. The only evidence of its existence is a small, crudely painted sign next to the door.

SCP-2352 is capable of communicating with human speech, however if one enters the room, it will not begin any conversation, but instead will begin a series of nonsensical, often nonsensical, sentences. This will often lead to the phrase 'I'm not a bad person', followed by what appears to be a blank, unreadable image of what appears to be a large, heavily-trailed train. Occasionally, this train will approach a house and the door will open.

SCP-2352's anomalous properties begin to manifest if the subject, or a subject, is in the room of SCP-2352 and takes any action on the subject, such as moving, running, or talking. If a subject takes any action on the subject, the words on the subject's face will begin to turn blue.

A.3 Selected generated samples 2

Note that the bold text is the prompt. The last truncated sentence is removed manually.

This is an SCP-Foundation fiction:

Title: The Most Perfect Way

Item : SCP-4850

Object Class: Safe

Special Containment Procedures: SCP-4850 is to be stored in a secure locker at Site-27. No individuals or personnel are to be given access to SCP-4850. SCP-4850 is to be worn only by personnel with a security clearance of 3/4850/1.

Description: SCP-4850 is a tuxedo uniform of a size 7, worn by personnel who are not already in possession of a full size tuxedo. The item is in a state of disrepair and wears poorly, and should be stored in a plain white canvas tuxedo bag. No item has ever come into contact with SCP-4850, with the exception of a small scratch or a single mark or scratch or two on the back of the uniform. There is no identifying mark or scratch or two on the back of the uniform.

When SCP-4850 is worn on its own, the individual who wears it is immediately aware of all information regarding their own body and the details of what happened to it. It does not need to be worn on its own to be noticed by this effect. The individual also knows that their body has been disassembled into an object that has been made to resemble their own, and can no longer perceive the person who made the disassembly. This effect has no effect when worn on the body of another human, or when worn on another human, and is not affected by this effect.

It was a uniform worn by a senior researcher, and was purchased from a company with a reputation for quality.

Addendum 4850.1: Investigation

In order to analyze the effects of SCP-4850, a database of items with this tag were created, and scanned through it.