**Story Creation / Text Generation**

**Capstone 3**

**Dawn Betzel**

****

### **Introduction**

Several of us have had a lifelong love of reading. I am one of those people to whom a book is one of my favorite pastimes. In an earlier course, one of the projects that I worked on was a “Script Generator”. The project was to create a neural network to train a model on many Simpson’s scripts and then try to generate a new script. This created a spark for me, and I had always wanted to see if I could train a model on several novels and create a new story using text generation.

# **Data**

I was lucky enough to find several datasets to test on this project. It was fortunate that I found so many as there are some formats that worked better for training than others. I first started with the data at this kaggle site which contains sixteen novels by Stephen King, <https://www.kaggle.com/ttalbitt/stephen-king-books>. The novels were web scraped from the site Novel80. Using this data I achieved some success in generating a story, but it was missing all punctuation and did not deliver great results because of that I think. I also found another kaggle site <https://www.kaggle.com/santiviquez/hp1txt>. This one had the first Harry Potter novel, which when it trains on the model makes a very distinct story as the elements in a Harry Potter novel are different from most other types of writing. I also tried a few novels from <https://www.gutenberg.org/> which focuses on older works for which U.S. copyright has expired. I tried a couple of Sherlock Holmes novels with some nice results. Finally the one that I liked the most for this project was <https://www.kaggle.com/khulasasndh/game-of-thrones-books>. This is five Game Of Thrones books and it worked well again because the format was nice, there were five novels that I could combine and train on and the story is rather different from other types of writing.

### **Modelling**

As I said I started out with the idea of working on this project by starting with a project from another class. That was a project that used a Recurrent Neural Network (RNN) to train a model on the text file provided. One of the first problems that I encountered was that this notebook was from a class that I took years ago and it required an older version of TensorFlow. I did not want to revert to an older version of TensorFlow. I continued to try other tutorials that I was able to find using RNNs to train on text and generate new text, but the next issue that I encountered is that they were character based rather than word based. Using character based generation resulted in text that made very little sense.

In search of learning how to do word based text generation I found transfer learning using GPT2. I found one tutorial which used Fairy Tales to train on the GPT2 model and then generated a very short excerpt of a story. Once I had tried this I looked to learn more and found the book *Transformer for Natural Language Processing* by Denis Rothman. The final program I used for this project is one that I learned how to do from this book and modified for my purposes.

### **Output**

After much experimentation I got the model to generate several different samples that were long enough to be considered a short story. Each sample had between 4,000 and 7,000 word counts. The average short story should run anywhere from 5,000 to 10,000 words, but they can be anything above 1,000 words. I am including shortened samples following this paragraph to allow you to get an idea of what was generated without making this report far too long and unwieldy.

After training with five Game Of Thrones novels this is a portion of the sample generated:

A cold war had broken out. The sword had traveled with it and then had to be sent somewhere, somewhere, to strike down the foe. In two hours it had pierced the Iron Throne, and after six hours of war there was no man there for Robert III to beat down. "You are no longer a prisoner of the Dragon. Behold, I will hold all power, but my lords! You cannot call such a thing , you can say it. There are many who do the blood-and-spewing of our people ," thundered Robb, glancing over the ragged wound. Jaime Lannister, holding his arm as a blade, followed suit and Jaime stared deep into Theon's eye . The blade flew up to his head, glancing straight at him, "The traitor, the thief, the drunkard, I do not mean them to kill. They will not let him have a chance against my castle. I will hold them all. " "If the Dragon can not slay you, the Fire King has no time for you. You are my brothers," he declared wryly, in a voice that sounded like a jape as Robert turned, arms crossed, he saw through Cersei's long black hair, "but this man has no way of ending my day in his own place. I shall see to it that he kills or at least puts him back to our land. When I die my fate is sealed ." The old dragon eyed Robb . . . There has been a long delay. "Tell ye the truth, Robert," cried Davos, glancing at the sword the hanger was held in the holdfast. "You have an idea. You must see to it that I have held his head ." Ser Catelyn Stark had held one more , more —"I will save you my fellow-races," said his father as they looked up, and he looked down "there's no escaping it. It is the death of an old King, a young Stark, and you know well how dangerous its destiny is ." Jaime sat up, with Robert at his side as the two knights drove off the man.

The next excerpt is a sample after I had trained the model with several Sherlock Holmes texts:

But Mr. Morgan's face fell on Sherlock's. He spoke in the very tone he had never in his life expressed before: "No. No, no, not before I see my mother, the doctor, or a friend. Do not think at all of that, I assure you I will never do it myself again." In a strange way he replied like this; but his voice was firm but he did not want to sound so. When I sat down at the table, I would not stand down for a day in the hospital. I would not let his mind wander off to things like that.

Sherlock and his mother were now standing there listening for anything that Sherlock could come and tell them about to go on or not because they found it necessary to make such a stand before the doctor could be permitted a step if he looked up from his work. I did not give any consideration to the two friends, and they stood still watching the Doctor's face as Holmes got in.

In all this this was one of the most extraordinary episodes of my life and yet of mine. I never had such an ordinary childhood in my life without being in the home, with my husband and my little sister, and knowing little things, and in the presence of those who were my teachers and nurses. One morning when we had gone to St. James, and the Doctor and the two of us were out to dinner with our daughter and the grand-daughter of one of them, they found that her grandmother, sitting in my room, was standing by herself talking about her grandchildren and my grandparents only.

One final sample excerpt where I trained the model using a collection of Fairy Tales:

Once upon a time there lived an old man who came to seek

his fortune in beauty and wealth. He gave his old mistress the

wonderful portrait that now stood in his stead.

He therefore entrusted the request which she now gave him. He made a

charity of money to whoever could give it to him, and placed

her on his journeymen, and thus far he never made any mistake.

"Wherefore, my child?" said the old man.

"Well, it is all right, my child, that you have settled in home for

all eternity. I have, by the hand of my beloved, replaced the old

olde house with a new one. I want to give you my first diamond and

silver. For I am well satisfied with what I have obtained from you. You

must now purchase both of these things.

You can see that from these small excerpts that each sample is similar in writing style to what the model trained on. Although I am not providing the text I used for training for brevity, I can state that each sample resembles the format of the training text. I did not change them when I cut and pasted them into this report and you can see that the font on each sample is slightly different as well as the formatting. These adhere closer to the training text. I didn’t see the differences so well when I was running them in the Jupyter Notebooks, but when I pasted them in here I did note it.

### **Challenges and Further Work**

In working on this project as I stated it was my goal to see if I could find a way to generate text and create a new story similar to the novel or novels I was using to train the model. That task was difficult enough that I did not think much past that. Over the months I have worked on this project and especially as I was able to come close to my goal, I started to see more challenges that needed to be overcome. For instance once I was able to generate a small sample of text, say 500 to 1,000 words in length, how then should I continue on to get at least enough for a short story. There would need to be cohesion between the first sample generated and the next sample generated. I was informed by my mentor that he would hold on to the text generated when he needed to do this and reuse it as the prompt for his model. I did that with this code snippet:

raw\_text = input("Model prompt >>> ")

whileCount = 0

while whileCount <=10:

while not raw\_text:

print('Prompt should not be empty!')

raw\_text = input("Model prompt >>> ")

context\_tokens = enc.encode(raw\_text)

generated = 0

for \_ in range(nsamples // batch\_size):

out = sess.run(output, feed\_dict={

context: [context\_tokens for \_ in range(batch\_size)]

})[:, len(context\_tokens):]

for i in range(batch\_size):

generated += 1

text = enc.decode(out[i])

print(text)

raw\_text = text

whileCount = whileCount + 1

The next challenge that I encountered was that the sample generated would quite often end somewhere in the middle of a sentence. It would not be very likely that it would end at an appropriate punctuation mark. I created the following code snippet to ensure that my text samples would always end with a period.

text = ' softly of the age of heroes and the children of the forest. He had taken off'

textTruncated = ''

lenText = len(text)

i = 0

for c in reversed(text):

i = i + 1

if c == '.':

print(i)

textTruncated = text[0:lenText-(i-1)]

print(textTruncated)

Break

This particular project could turn into a never ending task if I let it. When I saw that I had what looked like closure with a period at the end of sentence, I wondered if perhaps I should have looked for other punctuation marks. I checked my reading history and it seemed to me that most stories end on a softer note with a statement. You would not want to leave a reader with an exclamation or a question would you? I then started pulling books out of my bookshelf to randomly check how they ended. I was correct in that the books in my keeping all did end on a statement, quite often ending with a period. Sometimes they would end with a direct quote by a character, so should I have checked for the period and double quotation as well. Maybe I should have but with further research I find the problem goes even deeper. Writing is an art form, and it is possible for stories to end in all different ways. There are even some stories that ended mid sentence, with no ending punctuation. Even though I have not yet found an example of such, I am also sure someone has tried ending a story with a question or exclamation. That gives rise to thought of how you would decide on an ending point for a story. I have not yet gotten to a good enough answer for all stories, but I believe it lies somewhere in you may have to think what genre the writing occupies. Novels dealing with relationships seem more likely to end with a quote from a character, action novels, horror novels and science fiction seem to end more often with a statement. This is just from research using my very limited bookshelves.

Another challenge that I can see from thinking my way through this as I was working on this. All stories have some sort of title that relates to the story. A challenge for the future is how to generate a title for a given story. Would you do it at the beginning and then try to generate a story based on that, perhaps letting the person who runs the model enter in a title, a genre and a beginning block of text to start the model? Doing some research in this area has told me that most story titles come later in the process and can even change as the work on the story is ongoing. That would seem to suggest perhaps that it might be better to wait until the story is complete and then perhaps do some analysis on it to create a title.

Were you to want to take this from a short story to something longer like a novella or novel it would get even more complicated. Then you would have to worry about where to place chapter breaks. How would you determine the beginning and end to a chapter? How long should each chapter be? If you look at some authors and books you will see that each author has a certain cadence. They tend to write chapters most often either longer or some authors keep their chapter size fairly small. Of course you can see from the output I am currently getting in the samples that the output tends to look like the text used for training. Maybe the best way to progress in trying to do a longer story would be to get better input material.

This leads me to the last challenge I have considered. I want to keep working on this project and when I do I plan to do some web scraping and see if I can get different formats to test my theories on what the model might do when given different input. I have looked at a couple of tutorials for web scraping and think that will be the next thing I try to do.

### **Conclusion**

Natural Language Generation (NLG) or Text Generation is a subfield of Natural Language Processing (NLP). Its goal is to generate meaningful phrases and sentences in the form of human-written text. It can be used in a wide range of use cases from writing long form content like reports, articles, product descriptions, social media posts. I chose to focus exclusively on stories, but this project could easily have been switched to another area such as training the model on new other writings such as articles or reports and generating new articles or reports.

NLG fascinated me five years ago when I first ran through a class and learned to use it. It was slower back then and consumed more resources and didn’t in that project generate very well written things. Learning from my mentor about transfer learning and seeing how much quicker and easier it made doing text generation makes me think that this is an area I would expect to grow in the future and see more and more things developed that use NLG.

Finally in my thinking through all the different things I would need to try to generate a longer story, I can see there are also many non-trivial challenges that will come with growth in this area of Natural Language Processing. Trying to find the right way and tone to start and end things is rather difficult and goes to the nature of the creative process. That is part of what attracts me to this area of machine learning so much is that there are so many interesting challenges to think through.

### **References**

Rothman, D (2021) *Transformers for Natural Language Processing*

Musk, E (2021) [Elon Musk AI Text Generator with LSTMs in Tensorflow 2 (analyticsvidhya.com)](https://www.analyticsvidhya.com/blog/2020/10/elon-musk-ai-text-generator-with-lstms-in-tensorflow-2/)

TensorFlow Resources <https://www.tensorflow.org/text/tutorials/text_generation>

Phan, D (2021) [*www.twilio.com/blog/ghost-writer-spooky-task-queues-python-openai-gpt3*](http://www.twilio.com/blog/ghost-writer-spooky-task-queues-python-openai-gpt3)

Tucker, A (2021) [*https://www.kaggle.com/tuckerarrants/text-generation-with-huggingface-gpt2*](https://www.kaggle.com/tuckerarrants/text-generation-with-huggingface-gpt2)