Fall 2021

Fine-tuning Daria: Exploring the Implications of Temperature, Epochs, & Corpus Size on GPT-2 Screenplay Generation

Rebecca Lawson
Kenyon College, lawson1@kenyon.edu

Recommended Citation

Lawson, Rebecca, "Fine-tuning Daria: Exploring the Implications of Temperature, Epochs, & Corpus Size on GPT-2 Screenplay Generation" (2021). *IPHS 300: Artificial Intelligence for the Humanities: Text, Image, and Sound.* Paper 29.
https://digital.kenyon.edu/dh_iphs_ai/29

Finetuning *Daria*:

Exploring the Implications of Temperature, Epochs, & Corpus Size on GPT-2 Screenplay Generation

Rebecca Lawson,

IPHS 300 - AI for the Humanities, Professors Elkins and Chun, Kenyon College

Introduction

GPT-2 has a number of hyper parameters that can be tuned to adjust the generated text, including temperature, number of epochs, size of the training text corpus, batch size, and number of samples generated. Since GPT-2 is a new AI model, the effects of fine-tuning are still relatively unknown and unexplored. This paper summits experiments adjusting these GPT-2 parameters and how they affect the text output that the model generates.

Background

GPT-2 is a large Transformer model created by OpenAI as a general-purpose learner whose aim is to predict the next item in any arbitrary sequence. It has been trained on the text content of 8 million web pages in order to learn how to compose text. It can be fine-tuned on a specified corpus of text and generate similar text, attempting to learn the style and content of the fine-tuning corpus and imitating the style and content of it. GPT-2 can continue writing from an assigned starting prompt or generate an entirely new piece of text. It can also summarize, translate, and answer questions about a text it was fed.

Methodology

GPT-2's genesis as a general-purpose learner makes it extremely versatile in its applications. It can generate any type of text it can be fed: novels, lyrics, poetry, news reports, journalistic articles, scripts–you name it. I trained GPT-2 on all 5 seasons of the MTV cartoon series *Daria* in order to assess its ability to compose television scripts, or screenplays. While anyone can feed GPT-2 a corpus of episodes of a television series and receive an output of text, I am interested in determining what hyperparameter settings yield the best TV screenplays. I will be experimenting with fine-tuning the temperature, number of epochs, and corpus size to find the optimal settings for composing an episode of *Daria*. I tested temperatures of 0.5, 0.7, and 0.9, epoch levels of 500, 1000, 2500, and 5000 epochs, and corpus sizes of approximately 285,00, 142,000, and 28,500 tokens. I used a temperature of 0.7, 1000 epochs, and the full corpus of 285,000 tokens as the default settings when testing each variable.

Results & Analysis

TEMPERATURE

The first variable explored was the temperature of the model. The temperature value affects the probability distribution that GPT-2 uses to determine the next word in the sequence it is generating. When the temperature is high, the probabilities of the possible words to come next become closer in value. Thus, with high temperature values, all words have a similar probability of being the next word, which means they depend less on the previous words in the sample and on the lexical makeup of the training corpus for their probability value. So, we expect to have more randomness and less cohesion from word-to-word in samples with high temperature. Low temperature values follow a similar logic–when the temperature is low, the probabilities of the possible words to come next have much more variability. This means they depend more on the previous words in the sample and on the lexical makeup of the training corpus, so we expect much more cohesion from word-to-word and more parallels with the training corpus. But what exactly constitutes a "high" or "low" temperature? And which level works best for generating a screenplay? To answer this, I ran a GPT-2 model on *Daria* (the entire series) three times and had it generate a corpus of one hundred five-hundred-word samples. Each time, I set the temperature to a different value–using 0.7 as a baseline of sorts, I ran the model with a temperature of 0.5, a temperature of 0.7, and a temperature of 0.9. The metrics I used to judge the efficacy of each temperature level were repetition, or the presence of one line repeated three or more times; topic range/cohesion, or how varied the topics were within a sample and how far they fell from the typical topics within the *Daria* universe; and level of invention, or how often GPT-2 creates new characters or, in some cases, new words.

With a temperature of 0.5, the generated samples were not great. 72 out of the 100 samples had noticeable repetition(~75%), many to the point where it totally detracted from the legibility of the sample. The topic range of the samples was very narrow–most samples only broached one topic, if they broached any at all. The excessive repetition of many samples made it hard for a topic to be brought up, but if one was, such as a test or a necklace, it was usually the only one in the sample. The samples consisted mainly of vague statements, such as "Yeah, I know you, but I just don't feel like it" and "I'm sure you'll find it interesting," which do not contribute much towards the development of a topic. The samples contained no new characters, and no GPT-2-invented words. There was actually a lack of character diversity: the samples almost exclusively featured only the five most prevalent characters, and a lot of side characters were completely absent.

With a temperature of 0.7, the generated samples were markedly better than those with a temperature of 0.5. Only 9 out of the 100 samples featured noticeable repetition (~10%), and the repetition did not detract from the legibility of the sample as severely. The topic range was wider–each sample had at least one distinct topic. The samples with multiple topics did not do much to facilitate the transition from one to another, but each topic was developed for a couple lines before it switched. Additionally, GPT-2 only invented a new character in one of the 100 samples: someone named Gabe. These samples also had much more character diversity, featuring many of the more minor characters. No new words were invented in any of these samples, so I would say that the level of invention was pretty low, but the diversity of existing characters and words was much higher than that of the lower-temperature samples.

With a temperature of 0.9, the generated samples were kind of crazy. Only 2 out of the 100 samples featured noticeable repetition (~2%). The topic range was huge–most samples touched on at least 4 different topics. Again, these topics were not very sensically-grouped and there were no transitions between them. The character diversity, regarding existing characters, was pretty good–a lot of the minor characters showed up throughout the samples. The samples even featured multiple new characters that GPT-2 invented, such as "Brutus," "Hunter S. Thompson," "Lionspawn," and "Lambert." GPT-2 also invented new words, like "mayhemer," "duffy-druffy," "thumbsticks," "coochie-eyed," "storytellerical," and "strook." Thus, the lexical diversity was way above that of the lower-temperature samples, which only featured existing words.

From my analysis of the GPT-2 generated samples at different temperatures, I have determined that temperature significantly affects the level of repetition, topic range, and lexical diversity of the text output. All of these make sense, as temperature has to do with the probability distribution of words that the model has to choose from when determining the next word in a sample. Lower temperature samples have much higher repetition because only the words most prevalent in the training text have high probability values and other words that are less prevalent in or absent from the training text have extremely low probability values. Low temperature probability distributions also cause low topic and lexical diversity by this same logic. Higher temperature samples have almost no repetition, but high topic and lexical diversity. This is because many words have similar probabilities of being the next word in the sequence, so a word's chance of showing up is much less dependent on the prevalence of the word in the training text. Therefore, we have much more randomness in the distribution of words in samples with high temperatures. With screenplay writing, I found that the sweet spot for temperature is at about 0.7. I would even consider tuning it down a tiny bit more, like maybe to 0.67-ish, in hopes of finding a bit more topic cohesion. This would likely come at the cost of a bit more repetition within the samples, but that could be easily taken out by human editors. Models with temperatures of 0.5 have way too much repetition and too few topics to compose an enjoyable episode, and models with temperatures of 0.9 generate too many disconnected topics to make a coherent episode.

NUMBER OF EPOCHS

The next variable I am exploring is the number of epochs that the model is trained over. This refers to the amount of times that the model goes over, or reviews, the training text it is fed in order to "learn" it so that it can generate new text in its likeness. I compare models trained over 500 epochs, 1000 epochs, 2500 epochs, and 5000 epochs in order to determine how the number of epochs affects the text output. The metrics I am using to judge the efficacy of each model are repetition, which, again, I define as occurring when a line is repeated three or more times; the number of odd starts, which I define as when a sample starts with some type of writing that is not in the style of a screenplay or completely unrelated to *Daria*; and the presence of overfitting, or when the model generates text identical to that in the training text.

The model trained over 500 epochs definitely had some issues. Approximately 50% of the samples had repetition in them. While it usually did not detract from the sense of the sample, it still was obvious and redundant enough that it would not make it past the writers' room for a television show. And, with half the samples containing repetition of some sort, it is hard to ignore and would be time-consuming to manually correct. This model also did poorly with odd starts–approximately 33% of the samples it generated started with some type of writing that was definitely not *Daria*-eqsue. I refer to them as odd starts because within each sample, the model usually worked its way back to the script-style of *Daria* with the show's characters. A couple examples of odd starts are "WWE.com - The Ultimate Guide To The Best Of WWE.<|endoftext|>(CNN) -- As the sun rises, the first thing that begins to fade is the sun's love…" and "PSA: Do not use this picture for personal gain. The person in the picture is a college student." These appear to be instances where GPT-2 starts generating text inspired by some other corpus than the training text, then remembers that it is supposed to be writing a *Daria* episode, and tries to correct it as it generates. With 500 epochs of training, these occurred in one-third of the samples, which is not great when the goal is to be writing a screenplay. There was no evidence of overfitting in this model's output text.

The model trained over 1000 epochs improved upon some of the issues present with the 500 epoch model. Repetition was present in only approximately 10% of the samples, and it did not tend to detract from the sense of the samples. The little repetition that there was could be corrected by human writers without much difficulty. Only about 22% of the samples contained odd starts, which is an improvement from the previous model. Again, there was no evidence of overfitting with this model.

The model trained over 2500 epochs had repetition present in only about 8% of the samples it generated. Again, the repetition did not ruin the samples it was in and could be easily removed by humans. About 6% of the samples contained odd starts, which is a significant improvement from the 1000 epoch model. This shows that as the number of epochs increases, the model learns the style and form of the training text better and makes more accurate guesses about how samples should start. The 2500 epoch model also had no instances of overfitting.

The model trained over 5000 epochs had the least repetition present–only about 2% of the samples it generated contained repetition. Only 1 of the 100 samples generated had an odd start, which shows that the model has almost perfectly learned the structure of a *Daria* episode. But, this model, unlike the other models, was prone to overfitting. There were 3 noticeable instances of overfitting where chunks of lines were copied straight out of the training text. There may have been even more overfitting, especially with single lines, which are less noticeable. Thus, I found that there is such a thing as over-training a model, and that there is an upper limit to the amount of training epochs in order to avoid overfitting.

Ultimately, I found that increasing the number of epochs a model is trained on improves the performance of the model up to a point. The optimal epoch setting appears to be around 2500 epochs, though there is room for closer testing on where exactly the overfitting threshold lies between 2500 and 5000 epochs. The 2500 epoch model had minimal repetition and odd starts and no instances of overfitting, making it the best-performing model out of the four.

CORPUS SIZE

The last variable I am testing is the size of the corpus that the model is fed. My full training corpus, containing all 5 seasons of *Daria*, consisted of approximately 285,000 tokens (or words). I trained a model on the full training corpus, one on half the corpus and one on 10% of the corpus. The metrics I used to judge the effects of corpus size were repetition, defined as before; odd starts, as previously defined; and the presence of overfitting.

The model trained on 100% of the training corpus had repetition in about 10% of its samples. Again, the repetition was not extremely disruptive to the flow of the samples it appeared in, but it would be nice to avoid. Odd starts were present in about 22% of the generated samples for this model, which is not terrible, but it shows that the model has not fully learned the structure of a *Daria* episode. There were no discernible instances of overfitting in this model's generated text samples.

The model trained on 50% of the training corpus, which contained about 142,000 words or tokens, featured repetition in approximately 22% of its samples, which is more than the full-corpus model. Additionally, about 23% of this model's samples had odd starts, which is similar to the number of odd starts the full corpus model generated. There were no noticeable incidences of overfitting in the half-corpus model either.

The third model, trained on 10% of the training corpus or approximately 28,500 words, had a much smaller amount of repetition than the half-corpus model: about 5% of the samples. Thus, we see that while repetition's correlation with corpus size is not always negative–there appears to be a threshold somewhere between 142,000 and 28,500 tokens where it becomes positive again. Odd starts were present in roughly 20% of the generated samples, which again is similar to the other two models. This model's output text was rife with overfitting: at the least, 60% of the samples featured lines or chunks of lines ripped right out of the training text. I say "at least" here because overfitting is difficult to exhaustively account for unless you have the entire series of *Daria* memorized. Regardless, there was a ton of overfitting with this model. This overfitting may have contributed to the lack of repetition within this model's samples, as there is not much repetition in the training corpus it heavily lifted from.

All in all, corpus size seemed to have the most effect on overfitting: the 100% corpus size and the 50% corpus size were large enough to avoid overfitting, but the 10% corpus size model fell prey to heavy overfitting. Thus, I would recommend not using training corpuses smaller than 140,000 tokens in order to avoid overfitting.

Tabled Results

Metrics for Temperature Levels

Temperature	0.5	0.7	0.9
Repetition	72%	9%	2%
Topic Range	narrow	medium	broad
Diversity/ Invention	little	fair	extreme

Metrics for Epoch Levels

# of Epochs	500	1000	2500	5000
Repetition	50%	9%	8%	2%
Odd Starts	33%	22%	6%	1%
Overfitting	none	none	none	3%

Metrics for Corpus Sizes

Corpus Size	100%	50%	10%
Repetition	9%	22%	5%
Odd Starts	22%	23%	20%
Overfitting	none	none	60%

Conclusion

Ultimately, I found that each of the parameters–temperature, number of epochs, and corpus size–does have a significant impact on the quality of the screenplay the model generates. Temperature was the most powerful parameter, but all three were important for creating legible and original scenes. The most effective setting was trained with a temperature of 0.7, 2500 epochs, and the full corpus size (285,000 tokens). While this setting generated the least repetitive non-overfitted text with the least odd starts, the content of the generated text was still far from that of an episode of *Daria* that would make it our of the writers' room. The topic range was appropriate, but the samples did not make much sense on a sentence-to-sentence basis and lacked character continuity. Thus, I conclude that, even with ideal parameter settings, GPT-2 still has a ways to go before generating screenplays at the level of human writers.

Bibliography

Isozaki, Isamu. "Understanding the GPT-2 Source Code Part 1." Medium, Analytics Vidhya, 8 Nov. 2019, medium.com/analytics-vidhya/understanding-the-gpt-2-source-code-part-1-4481328ee10b.

Radford, Alec. "Better Language Models and Their Implications." OpenAI, OpenAI, 3 May 2021, openai.com/blog/better-language-models/.