Title: Leveraging GPT-2 for Text Generation on Drake Lyrics

Github link:
https://github.com/dylansaunders23/DeepLearning/blob/main/DeepLearningProject.ipynb

---

Introduction:

In this project, we delve into the realm of text generation using the OpenAI's GPT-2 language model. We attempted to emulate the lyrical style of the artist Drake. By leveraging the transformative power of large-scale transformer models, we aim to capture the essence of Drake's lyricism and create text that resonates with his audience. This endeavor involves preprocessing lyrics datasets, fine-tuning the GPT-2 model, and evaluating its performance in generating compelling and contextually relevant text.

Previous Solutions:

Historically, text generation tasks have been approached using rule-based systems or statistical models with handcrafted features. However, these methods often fell short in capturing the intricacies of creative expression, particularly in the domain of lyric generation. The advent of transformer-based models like GPT-2 has revolutionized natural language processing by employing self-attention mechanisms to capture long-range dependencies in text data. This enables the generation of high-quality and contextually relevant text, setting new benchmarks in text generation tasks.

In previous submissions, we loaded the data from Kaggle, and then put it through a series of preprocessing and tokenization steps. This included removing the stopwords, cutting out any extraneous characters, and splitting the data into train, test, and validation splits. We then attempted to design our own model using the Keras library in TensorFlow, however, did so to little/no avail (we were hovering around 2% accuracy). We paired what we learned on our LLMs-focused assessment with our feedback from Professor Gyires-Tóth (generate examples, introduce more advanced models, implement text generation with temperature) and ultimately settled on pivoting our approach to using OpenAI's GPT-2 for text processing, and then fine-tuning the model to suit our specific needs.

Dataset:

1. Drake Lyrics Dataset: This dataset contains lyrics from over 260 songs by Drake, providing a rich source of textual data for training our model. Each entry in the dataset represents the lyrical content of a specific song.
2. Other Lyrics Dataset: In addition to Drake's lyrics, we utilize a dataset comprising lyrics from various artists. This dataset serves as a comparative reference, allowing us to train and evaluate the GPT-2 model's ability to distinguish Drake's lyrical style from that of other artists.
3. In the data processing phase, we combined the Drake Lyrics Dataset with lyrics from various artists to create a comprehensive training set. This combined dataset consists of 80% Drake songs and 20% songs from other artists, ensuring a diverse representation of lyrical styles. By incorporating lyrics from different sources, we enrich the model's training data, enabling it to capture a broader range of linguistic patterns and nuances inherent in rap and hip-hop music. This approach enhances the model's ability to generate text that resonates with Drake's lyrical style while also allowing for comparison and differentiation from other artists' styles.

Proposed Method:

1. Data Preprocessing: We preprocess the lyrics data by removing HTML tags, punctuation, and other non-essential characters. Additionally, we filter out metadata and extraneous information to focus exclusively on the textual content of the lyrics.
2. Fine-tuning GPT-2: Leveraging the Hugging Face Transformers library, we fine-tune a pre-trained GPT-2 model on the Drake Lyrics Dataset. Fine-tuning involves adjusting the model's parameters to the specific task of generating Drake-like lyrics, thereby enhancing its capacity to capture the nuances of Drake's lyrical style.
3. Training and Evaluation: Throughout the training process, we monitor key metrics such as training loss, validation loss, and early stopping criteria. Employing a custom training loop and early stopping based on validation loss, we aim to prevent overfitting and improve the model's generalization performance.

Evaluation Method:

In evaluating the performance of our fine-tuned GPT-2 model, we consider several metrics:

- Training Loss: This metric provides insight into the model's performance on the training dataset, serving as a measure of its learning progress.
- Validation Loss: Calculated on a separate validation dataset, validation loss indicates the model's ability to generalize to unseen data and maintain coherence.
- Early Stopping: By monitoring the trend of validation loss during training, we implement early stopping to prevent overfitting and ensure the model's robustness.
- Text Generation: After the model has been trained, we generate some sample lyrics and compare these to examples from the dataset, and also our understanding of Drake as lifetime followers of both Drake and Rap/Hip-Hop.

Results:

After training, the model was able to generate Drake lyrics with relative success. The validation loss for the model was 3.16, and while the lyrics themselves generally don't make much sense, the tone and style is very much Drake. Below are some examples of prompts and the results from the model.

Prompt: "Ay! "

Result: Ay! // // Gyal bday and dont keep it just // Cause Im doin brann check // She gon make me break then // I got some gyal in the post

Prompt: Doing Deep Learning with my friends

Result: Doing Deep Learning with my friends // Ill be on ice teamed by yelle never leave the city fallin // You gotta know him up him on his friends

Discussion:

Our approach capitalizes on the inherent strengths of transformer-based models, particularly their parallelism and ability to capture long-range dependencies in text data. By fine-tuning GPT-2 on Drake's lyrics dataset, we tailor the model to emulate Drake's unique lyrical style, thus enhancing its ability to generate contextually relevant text. The selection of appropriate hyperparameters, such as batch size, block size, and evaluation strategy, plays a crucial role in optimizing the model's performance and ensuring efficient training. Furthermore, the inclusion of early stopping mechanisms helps mitigate the risk of overfitting, thereby improving the model's generalization capacity.

Conclusion:

In conclusion, our project demonstrates the effectiveness of leveraging GPT-2 for text generation tasks, with a specific focus on emulating Drake's lyrical style. Through meticulous preprocessing, fine-tuning, and evaluation, we showcase the model's capacity to produce compelling and contextually relevant text reminiscent of Drake's music. By harnessing the power of transformer-based models and incorporating domain-specific datasets and evaluation strategies, we pave the way for future advancements in creative text generation tasks.