Beatles Inspired Musical Generation Using a RNN and GPT-3

Garrett T. Birch
Data Science Undergraduate
University of California - San Diego
La Jolla, 92093
gbirch@ucsd.edu

Joseph Perez
Data Science Undergraduate
University of California - San Diego
La Jolla, 92093
jop010@ucsd.edu

Matin Ghaffari
Data Science Undergraduate
University of California - San Diego
La Jolla, 92093
mghaffari@ucsd.edu

Project Repository: https://github.com/gbirch11/LIGN167-MusicGenerator

1 Abstract

The use of a Character Recurrent Neural Networks (RNNs) and Generative Pretrained Transformer 3 (GPT-3) has shown promise in generating music. In this research, we explored the potential of these techniques in generating unique musical compositions. Through a series of experiments, we found that the use of Character RNNs and GPT-3 resulted in the generation of relatively diverse and original musical pieces. Additionally, we observed that the use of these techniques allowed for the incorporation of various musical styles and influences, resulting in a more rich and nuanced output. Our findings suggest that the use of Character RNNs and GPT-3 has the potential to enhance the creative process in music composition - especially with potentially allowing musicians legacies to live on further.

2 Introduction

The field of music has always been closely intertwined with technology, from the earliest musical instruments to the modern digital audio workstation. In recent years, the use of artificial intelligence (AI) has emerged as a promising tool for enhancing the creative process in music composition. One of the key techniques used in this context is the application of Character Recurrent Neural Networks (RNNs) and Generative Pretrained Transformer 3 (GPT-3). In this research paper, we explore the potential of these techniques in generating unique musical compositions.

Character RNNs and GPT-3 are machine learning algorithms that have been widely used in various natural language processing tasks. These algorithms use a large dataset of text to "learn" the patterns and structures of language, and are able to generate new text based on the input provided. In the context of music, these algorithms can be trained on a large dataset of musical compositions and then used to generate new music based on the input provided.

In this project, we aim to evaluate the effectiveness of using Character RNNs and GPT-3 in generating unique and original musical compositions. We will conduct a series of experiments to explore the potential for incorporating various musical styles and influences into the generated music. Additionally, we will analyze the characteristics of the generated music, such as melodic motion, and compare it to human-generated compositions.

Preprint. Under review.

The findings of this research will provide insight into the potential of using AI in music composition. This can potentially open up new creative possibilities for musicians and composers, as well as provide new avenues for musical exploration. Furthermore, the use of AI in music composition can also enhance the efficiency and productivity of the creative process, allowing for the generation of large volumes of music in a relatively short time.

3 Models

3.1 Character RNN for Lyric Generation

A character RNN is a type of machine learning algorithm that is widely used in natural language processing tasks. This algorithm uses a large dataset of text to "learn" the patterns and structures of language, and is able to generate new text based on the input provided. In the context of generating new Beatles lyrics, the character RNN was trained on a large dataset of Beatles lyrics.

More specifically, the specific kind of RNN that we implemented is LSTM (Long Short Term Memory). From our research we learned that LSTM (Long Short-Term Memory) will perform better for our sequence prediction task since it is very capable of learning long-term dependencies (unlike traditional RNNs) as it has a long term memory that is able to remember things selectively in addition to short term memory. Thus, we found that LSTM would be best suited for our task because it eliminates many of the concerns regarding the quality of long-term dependencies with traditional RNNs. The LSTM is able to mitigate issues with long-term dependencies thanks to it's clever design. The LSTM's design includes the cell state, which is the long-term memory since the recursive nature of the cells build long term dependencies of previous inputs' information which is remembered selectively via the forget and input gate. The short-term memory component is the hidden state, which stores previous immediate information. Thus long-term dependencies are learned thanks to do these memory states along with gates that allow for the model to remeber selectively. The forget gate uses a sigmoid function on the hidden state and the current input to estimate which features of the cell state should be forgotten. While the input gate updates the cell state (determines what is remembered in long term) by passing the previous hidden state and the current input into a sigmoid and tanh function which are then multiplied point-by-point. Lastly, the output gate determines the next hidden state (determines what is remembered in short term) by using a sigmoid function on the previous hidden state and the current input and tanh on the updated cell state which are then multiplied point-by-point [7, 1].

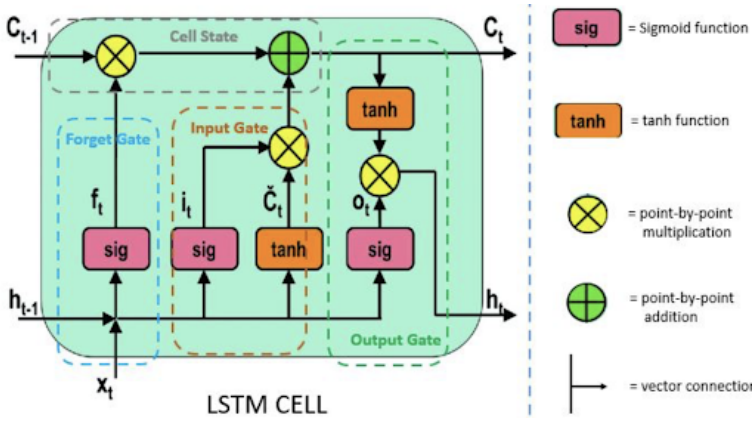

Figure 1: Diagram of Described LSTM model

Prior to training, preprocessing of our data was required in order to yield quality predictions. First, we preprocessed the data by removing all punctuation except commas, periods, exclamation points, and question marks as these are often important to the song structure. Additionally, we removed all newline characters (`\n`) and ensured that every song ended with `\n` as the special character for the stop token. Lastly, we ensured that if a song didn't end with a period or exclamation mark then a period was added as the last character of the song prior to the newline character. These prepocessing steps were important not only for eliminating errors often associated with irrelevant punctuation

marks, but these steps also significantly aid the model to learn song structure better especially with regards to endings.

Additionally, we conducted exploratory data analysis in order to gain insight into how we should design our model parameters. Our analysis shown in Figure 2 provided us with insight that we should be predicting from 700 - 1000 characters, since that is where the distribution of Beatles song length appears to be centered around. Additionally, we plotted the frequencies of the last character within the lyrics and learned that most songs end with a period, which prompted our decision to add a period to songs that didn't end with a period or exclamation mark.

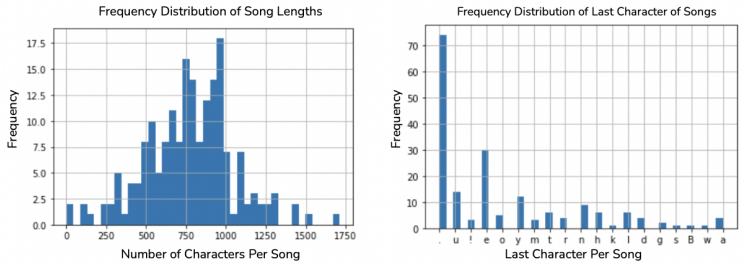

Figure 2: Exploratory Data Analysis used to make informed decisions about data prepossessing and model parameters.

The training process involved feeding the character RNN the lyrics of multiple Beatles songs, allowing the algorithm to "learn" the patterns and structures of the Beatles' lyrics. More specifically, we generate a Beatles song by training a RNN on all Beatles songs, using random batches, in order for our model to be able to predict a Beatles song when given some text as the starting seed of the lyrics. The model will pass the characters of the starting seed through the RNN. The hidden state and cell state of the last character in the seed will be used in order to predict the next character using the softmax function. This process of using the previous characters' hidden and cell state to predict the next character is repeated until the song is generated. Once the character RNN was trained, it was able to generate new lyrics based on the input provided. For example, if the input was the first few words of a Beatles song, the character RNN would generate the rest of the lyrics in the style of the Beatles. Ultimately, the generated songs will be between 700-1000 characters long and the model will stop predicting when either a the special stop token of a song is predicted (`\n`), or once it passes 700 characters and predicts a sentence ending such as a period or an exclamation mark, otherwise it will predict until the set maximum output-size of 1080 characters.

More specifically, for each epoch our model trains random batches of character sequences. We found the best performance when using a batch size of 128 and sequence length of 60. Furthermore we found the best performance when using only a single layer for our LSTM with a hidden size of 512. We use the Adam as our optimizer and we evaluate our loss using cross entropy. We found that our results best converge when using a learning rate of 0.001 and 10,000 epochs. The loss after 10,000 epochs was 0.15. Lastly we found that the best temperature for our softmax is 0.9. Temperatures that were too low made the model too confident in its predictions or temperatures that were too high made the model too unpredictable, resulting in errors.

Ultimately, the generated lyrics were then compared to the original lyrics to evaluate the accuracy and similarity of the generated lyrics. This allowed for the creation of new Beatles lyrics that were similar to the original lyrics in style and content. The use of a character RNN in generating new Beatles lyrics allowed for the creation of unique and original lyrics that retained the characteristic style of the Beatles. This process has the potential to enhance the creative process in music composition, allowing for the generation of large volumes of music in a relatively short time.

3.2 GPT-3 Fine Tuned Model

Generative Pretrained Transformer 3 (GPT-3) is a state-of-the-art natural language processing algorithm that has been widely used in various tasks, including generating music. In this context, we fine-tuned GPT-3 on a large dataset of Beatles songs, allowing the algorithm to "learn" the patterns and structures of the Beatles' music.

To start, our problem differed from the AI Tunes article in that we needed to start with MIDI data as that was the only available file option when dealing with data for a specific artist's catalog. We used the Lakh dataset of Beatles songs in MIDI format [1] as our dataset. Following a similar process as the previously mentioned article, we then needed to convert this MIDI data into a format that would retain the song's musicality and information such as tempo and measure length in text format which is why we converted the data into a XML format. XML format is highly structured and text heavy which means we then needed to convert this into a format that would work better for a predicted output. This is where we then convert the XML formatted data into ABC format instead using XML2ABC [2]. ABC format is a cleaner format with less structural text in the way which would allow for better outputs from our fine-tuned GPT3 model. ABC is a widely used format for representing musical notation that is easily readable by computers. Due to price constraints, we stuck with the less powerful and creatively capable Curie version of GPT3 for fine-tuning instead of Davinci. For fine-tuning, we trained the model on a prompt consisting of overhead information that includes the title of the song, key, and the lyrics. The completion target here was the songs in ABC format.

To generate musical notation in the form of ABC, we used a character RNN to generate lyrics that were then entered into the GPT-3 prompt. As mentioned prior, the character RNN was trained on a large dataset of Beatles lyrics, allowing it to "learn" the patterns and structures of the Beatles' lyrics. The idea here is that the generated music would learn some of this structure from the lyrics.

The generated musical notation in the ABC format was then used to create musical compositions. The generated music was then compared to the original music to evaluate the accuracy and similarity of the generated music. Here, the metric we used to evaluate the newly generated music to the original music was conjunct melodic motion. Conjunct melodic motion is when a phrase has motion where subsequent notes in a sequence move up or down by a semitone or a full tone. Greater than that would then be disjunct. We took the quantifiable average from the training set, and compared the newly generate songs against this average where the closer to the average is considered good.

The use of GPT-3 fine-tuned on Beatles songs, along with our character RNN for generating lyrics, allowed for the creation of unique and original music in the style of The Beatles where we were able to generate new single instrument songs that range from snippets that are less than a minute long to several minute long pieces.

4 Datasets

Various datasets were used to complete the tasks presented. Firstly, to train the RNN we used Beatles lyrics coming from the online community platform for data scientists and machine learning enthusiasts, Kaggle. The dataset contains all Beatles songs along with lyrics, album name, song name, composer, and year [3]. We simply just take the lyrics from this dataset for further preprocessing techniques. After the preprocessing was done on these lyrics that was mentioned in the Character RNN section, it was then sent to a text file for easy future use.

The next dataset was used was again from Kaggle but this time it contained MIDI files from the Lakh MIDI dataset [1]. This dataset contains a collection of 176,581 unique MIDI files for various different artists. Like the dataset before, we extract those songs out that were made and recorded by the Beatles to use as our training data for fine-tuning GPT-3. Because of this dataset being a collection of MIDI files, and our GPT prompt is attempting to generate ABC musical notation we have to go through a set of processing techniques of transforming the file format. The workflow looks like; MIDI -> XML -> ABC. As a result of this, we lose some information in the process but retain the necessities, such as; chords/notes, timing, artist, title, to generate our songs.

Overall, we would have ideally liked to use multiple artists spanning over numerous genres for our data - but this proved to be not feasible in the timeframe due to reading and analyzing all the MIDI files for fine-tuning GPT-3. Hence, we stuck with the Beatles as they are an iconic staple of the music industry and many people would like to hear their music carried on. This project could be tuned to include as many artists as any user would like, but keep in mind the computational cost.

[1] https://www.kaggle.com/datasets/imsparsh/lakh-midi-clean

[2] https://wim.vree.org/svgParse/xml2abc.html

[3] https://www.kaggle.com/datasets/yeonseokcho/beatles-lyrics

5 Results

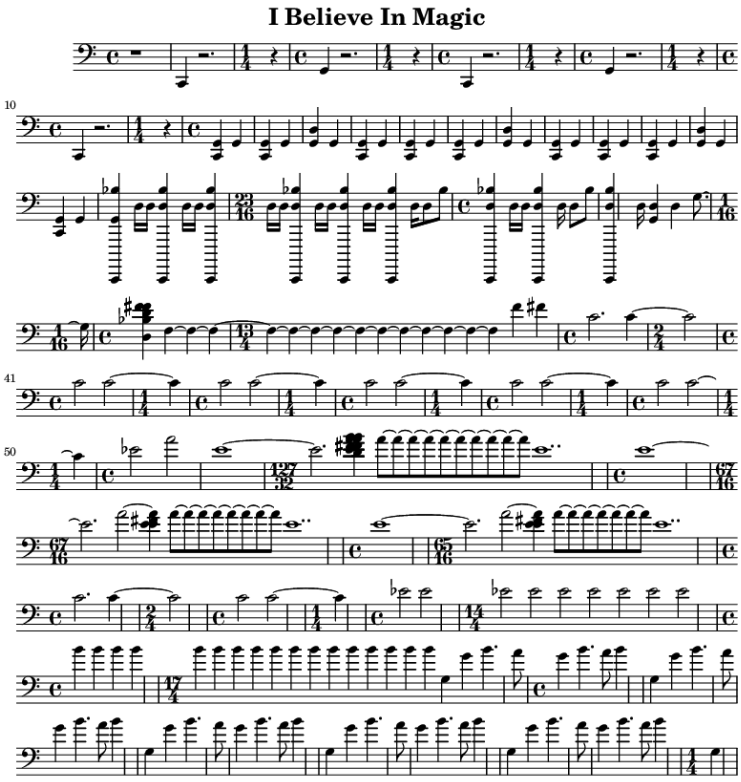

Figure 3: Example of Musical Notation Generated with starting seed "oh i believe in magic "

Overall, we were impressed that GPT-3 was able to be fined tuned to enable such task to be completed. The results of these musical pieces generated varied pretty heavily, some were very good while some lacked in the notes generated. We saw that the melodic notions between the two pieces were around the same when decent results were produced. However, we more rated these generated songs on how they sounded sonically to the human ear. We felt that this was the best option at the end of the day if the music sounds good, then its simply good music! A problem with this project is that about only one out of ten songs are decent and even fewer are pretty good; but it takes such minimal time to create these new generations that this is okay. Furthermore, this is raw data played through a MIDI player - not an actual guitarist giving it more emotion. We also aim more on the side of giving ideas to musicians, especially guitarists, for new riffs and ideas they can take away from the generated music. The use of a Digital Audio Workspace (DAW), like Reaper [4], is highly recommended to add effects, change certain parts of the music, and overall edit how you would like it.

The lyrics themselves were overall relatively good - however there were mistakes in the spelling that occurred every now and then. This is a result of using a Character RNN rather than one that predicts and generates words, the reasoning for choosing this is because it learns structure and rhyme schemes much better. Additionally, generating text at the character level allows for greater flexibility and creativity in the generated text, as the model has the ability to generate words that may not exist in the training data. This can lead to the generation of new and unique words and phrases that would not be possible with a word-level model. We also do not provide a set timing for when the lyrics should be sang given the generated song, it is up to the user/vocalist to decide.

Example Song Links;

1) I Believe in Magic generated using "oh i believe in magic " as starting seed.

2) Please Believe Me generated using "please believe me " as starting seed.

[4] https://www.reaper.fm/

3) Oh Yeah I Believe in Magic generated using "oh yeah i believe in magic" as starting seed.

6 Conclusion

In conclusion, we believe our project was a major success as we were able to generate decent sounding songs that had similar melodic motion to original Beatles' music. We were very impressed that GPT-3 was able to "learn" this musical notation through fine tuning and produce results that actually worked. With the future capabilities that large language models provide, like the new models for OpenAI, there is no doubt that soon musical generation will come into play more pronounced and have a lot more support. It will be very interesting to see what the future has to hold for the intersection of music and artificial intelligence.

For future use, we would like to fine tune on a better model than Curie; such as the latest Davinci model to hopefully produce more powerful, accurate results as this model is much bigger and overall better. We would also like to experiment around with OpenAI's newest release, ChatGPT [5], and see how well it can generate new musical notation for artists. More-so, the experimentation of Jukebox by OpenAi [6] would have been exciting to see how the actual inclusion of raw audio data to be trained on, showing the difference in actually analyzing audio vs text. It would be very interesting if we could create generations for a whole band, including; drums, bass guitar, lead guitar, rhytmn guitar, and lyrics/vocals and combine them all together to create a whole musical piece.

A huge shoutout to the following packages and applications that made this possible. Including; ChatGPT [5] for generating the rough draft version of our paper, the music21 package for python [4] enabling us to use this data, and finally AI-Tunes [2] and the article "Towards the Generation of Musical Explanations with GPT-3" [3] for inspiring our idea.

References

[1] Robert DiPietro and Gregory D. Hager. "Chapter 21 - Deep learning: RNNs and LSTM". In: *Handbook of Medical Image Computing and Computer Assisted Intervention*. Ed. by S. Kevin Zhou, Daniel Rueckert, and Gabor Fichtinger. The Elsevier and MICCAI Society Book Series. Academic Press, 2020, pp. 503–519. ISBN: 978-0-12-816176-0. DOI: https://doi.org/10.1016/B978-0-12-816176-0.00026-0. URL: https://www.sciencedirect.com/science/article/pii/B9780128161760000260.

[2] Robert A. Gonsalves. *AI-Tunes: Creating new songs with Artificial Intelligence*. Oct. 2021. URL: https://towardsdatascience.com/ai-tunes-creating-new-songs-with-artificial-intelligence-4fb383218146.

[3] Stephen James Krol, Maria Teresa Llano, and Jon McCormack. "Towards the Generation of Musical Explanations with GPT-3". In: (2022). DOI: 10.48550/ARXIV.2206.08264. URL: https://arxiv.org/abs/2206.08264.

[4] M.I.T. *music21: a toolkit for computer-aided musicology*. URL: http://web.mit.edu/music21/.

[5] OpenAI. *CHATGPT: Optimizing Language Models for Dialogue*. Nov. 2022. URL: https://openai.com/blog/chatgpt/.

[6] OpenAI. *Jukebox*. June 2021. URL: https://openai.com/blog/jukebox/.

[7] Gaurav Singhal. *Introduction to LSTM Units in RNN*. Sept. 2020. URL: https://www.pluralsight.com/guides/introduction-to-lstm-units-in-rnn.