NLP Final Project Report: Song Lyric Generator

Mirah Gordon, Henry Stachowiak, Albert Zhang

Artist Statement

Music is a great connector of people; its melodies, lyrics, and beats transcend the realm of conversation and can be enjoyed by everyone and can mold into any occasion of human experience. Current research even shows that humans "have a dedicated part of our brain for processing music," (Greater Good) supporting the idea that music is not only important superficially but functionally as well. The world of music, which has been around and celebrated for centuries, holds a deep connection to the evolution of spoken and written language as well; in fact evidence collected from fields of music cognition, language acquisition psychological development hypothesize that "language is thought of as a special type of music" (Psychology Today). It is through this lens that our group decided to delve into music and create our own musical art by combining the techniques and understanding of language via Natural Language Processing and the prevalence of music in the modern era.

The art seen as the culmination of this project is a song lyric generator which will take in song lyric prompts consisting of a line, a verse, song title, and artist and will output a continuation of the verse. Our hope in the completion of this project and the viewing of our art is that it will evoke the same emotional escape and connection as the original piece of music. A successful model will try to comprehend the intricacies of language we have discussed thoroughly over the course of this class and translate those nuances into a meaningful output: one that highlights the impact of the music. It has been interesting to explore this form of art, especially as it relies so heavily on our connection to language; exploring our data which contains all the segments of a single song (i.e. verse, chorus, bridge) and manipulating this by looking at common and uncommon words speaks to just how layered the notion of music is. Overall, our project is intended to build upon existing art in the form of music and utilize the ever-prevalent connection to language to find new meanings, new lyrics, and new understanding of our language.

Introduction

As touched on in the artist statement, we decided to pursue the first project option, which is to make an NLP powered piece of art. For this option, we're defining art as "the various branches of creative activity, such as painting, music, literature, and dance" (Oxford) and using this definition– we created a song lyric generator. In order to train our models to output new song verses, we needed a large amount of song lyric data to feed into the models; we procured this from a popular lyrical database (MusicOSet) and placed it in a dataframe in order to access all pertinent song data at once for the models to train from. The raw data included only two columns, a song id and the lyrics themselves. Other tools we employed during our cleaning and training were from standard Python libraries, including but not limited to: NLTK, Pandas, Numpy, Matplotlib and Seaborn, Sci-kit Learn, Tensorflow, and Keras.

Approach and Methodology

Our data was found online and is a compilation of song lyric data that has been specifically enhanced for music data mining; it was helpful to find such a feature rich dataset, as it already had labels in the lyrics for the different segments which make up a song. For example, the first song in the dataset, “thank you, next” by Ariana Grande, has lyric tags such as Pre-Chorus, Chorus, Verse 1, Verse 2, Bridge, etc. Even though the data set was already nicely processed, it still required additional cleaning and processing in order to prepare it for being fitted to the generative model. The cleaning process started by dropping any entries with empty values for the lyrics; the lyric text (which were strings) were then split by newlines and keys inside of the lyrics, which corresponded to the tags of song parts (see split_text function in final code). The final version of the dataset can be seen in Fig. 1 below.

id	song_id	lyrics	single_text
0	3e9HZxeyfWwjeyPAMmWSSQ	['[Verse 1]\nThought I\'d end up with Sean\nBu...	thank you next next \nthank you next next \nth...
1	5p7ujcrUXASCNwRaWNHR1C	["[Verse 1]\nFound you when your heart was bro...	tell me hows it feel sittin up there \nfeelin ...
2	2xLMifQCjDGFmkHkpNLD9h	['[Part I]\n\n[Intro: Drake]\nAstro, yeah\nSun...	woo made this here with all the ice on in the ...
4	1rqqCSm0Qe4I9rUvWncaom	["[Intro]\nHigh, high hopes\n\n[Chorus]\nHad t...	had to have high high hopes for a living \nsho...
5	0bYg9bo50gSsH3LtXe2SQn	["[Intro]\nI-I-I don't want a lot for Christma...	i dont want a lot for christmas \nthere is jus...

Figure 1. The head of the lyrics dataframe after all cleaning was completed

The last step in preprocessing the data was filtering for common and uncommon words in the lyrics; this was necessary for the training of our model so the output did not contain many one-off (few occurrence) words. We decided to set two thresholds here, one for minimum frequency of a word (set at 7) and one for a minimum sequence of lyrics (set at 5) and found that our dataset containing over 2 million words, saw 32,000 words with less than 7 appearances; this is a fairly insignificant portion of the data but knowing this information, we were ready to begin generating and training our model.

We chose to use two models and decided to run a comparison between them– one model which we built from scratch using TensorFlow libraries and one model we found online that is a text generator. The pre-made model was sourced from an online app and demo called InferKit, which is a fully operational user interface built upon their own neural network (InferKit). While the documentation for InferKit does not explicitly state what type of model is implemented on the backend, it accepts any form of text data (although up to 3000 characters) and outputs text data (up to 1000 characters); we assumed that this model would be trained on all text uploaded and thus have a far deeper reach of language and training sets, which could affect how musical or lyrical the output text was (in the context of songs).

Our model was built to be a recurrent neural network (RNN) and was inspired by a tutorial that we found from Active State (ActiveState). We chose to use this type of model because it specializes in recognizing patterns and is well known to be the standard for generative models. We also used the Long Short Term Memory variant with a word embedding layer. Word

embedding improves the speed of the model by representing the words in a dense way. The Long Short Term Memory variant of the RNN allows us to remember the important parts of the input sequence to improve the accuracy of our output. Our model was only trained on the data in the fully processed dataset, and as such only utilized song lyrics specifically, unlike the online text generator. This, likewise, could have affected how musical the output text was, as the training data had a much more narrow focus on song lyrics, although the variation in artist and genre might discount this.

Producing the Art

The final piece of art, a new set of song lyrics for a particular song, is produced solely from the output of our two models. Both models act in a very similar way to produce these lyrics (more detail on this is found in the above methodology section) but the actual lyrics do differ between them. In order to produce specific lyrics for this project, we created our own dataset of lyrics, which were called prompts, and consisted of 20 example songs with a single sentence of lyric text.

At each epoch iteration of our model, lyrics were generated and spit out which contained approximately 50 words in the new verse of the song (the final model was run with 2 epochs). The same prompts were entered into the pre-defined neural network and the text was similarly recorded and outputted as new song lyrics; this output contained only 200 characters, which makes comparing the two models slightly difficult but still gives enough basis to understand how well each model performed. Since the bulk of the work was done by creating our own model, the actual process of producing the art is quite simple, but relies heavily on the extensive training done to bring the model to fruition. The secondary model was far more shallow, as it only required uploading the prompt sentence into the text generator and recording the output. An example of the art (read: song lyrics) produced by the two models can be seen in Figures 2 and 3 (pre-defined model and our model, respectively), where in both cases the first line was the prompt and the following are the new lyrics.

```
You change your mind,
you will have to break up with me.
You can find someone to be your gal with all the money in the world."
Good God, this is not real life.
There is a woman I have liked for a while, but I am too
```

Figure 2. Final Art Generated from Defined Neural Network, using “Hot n Cold” by Katy Perry

```
----- Generating with seed:
"you change your mind"
you change your mind
 when i want you i dont wanna be me too
 to be my mama
 i dont know how we can do anything
 i do the same
 i got something to you
 you had a mess
 i can be the same
 my life
```

Figure 3. Final Art Generated from Our Neural Network, using “Hot n Cold” by Katy Perry

Results and Comparison

The results of our model were as follows. The model was able to output relatively usable lyrics. There were even lyrics that had a cohesive rhythm, such as “I don’t know how we can do anything // I do the same”. Additionally, there were some rhyming lyrics, but this rhyme was only achieved by ending the two lines with the same words. Overall, the lyrics were slightly nonsensical, but we were surprised about the musicality that the model was able to achieve with only two epochs of training.

There was an issue with our lyric generation, however. When the model ran into a word in the seed sentence that it did not recognize, it would crash. We tried a few separate solutions for this, the first one being decreasing the minimum threshold for a word to be included in the model to 4 from 7. However, this still caused issues, as some slang terms did not appear in enough music to reach the threshold. Another method that we attempted was to include the lyrics that we were using to seed the model with the actual training data, but that was not successful. The words did not reach our minimum appearance threshold of 7 to be included. We finally decided to specifically choose lyrics that we believed had enough appearances to reach the threshold. This is what we went with for our final model. Future additions to this model could attempt to replace words that do not appear often enough with “<UNK>”, similar to what we did in class.

Comparing this to the model we found and employed online from InferKit, we found that the inferkit model’s predictions made a lot more sense than our model’s predictions. The sentences were able to connect with each other and tell somewhat of a story. However, this model is heavily limited by the fact that it is a general purpose language prediction model, and not one specifically trained on lyrical data. Only a few of the outputs of the model were shaped like lyrics, with many being more novelistic than lyrical. Another output from this model was simply a list of different songs and their artists, possibly because the input was both a line from the song as well as the song itself. A line about food turned into a short story about a restaurant. To sum up, while the InferKit model produced more sensical results, our model produced more lyrical and song-like results.

Future Work and Improvements

While the models, data, and results of this project fit within the scope of a final project, there is so much more to be explored with our procured data and more song/lyric data in general. To start, it would be interesting to play around with further tuning the models we utilized, especially the neural network we built from scratch. Our model, a recurrent neural network (RNN), was based largely on a tutorial provided from Active State and was manipulated to fit our data; one of limitations we ran into when attempting to fine tune our model was the denseness of the model and the strain on our computers while training. While we had the bones of the model set up and ready to train, in order to improve accuracy of results in the model, we attempted to add an attention mechanism to the model, following further research (Medium). However, this proved to be too difficult given the time and computing constraints we faced, and ultimately we were unable to incorporate it into this iteration of the project.

In doing the project again or expanding upon what we have done so far, adding a full attention mechanism would most likely improve our results even further and create a more efficiently functioning model. This would most likely require far greater computing power than any of our project members have access to; an external GPU would present enough fuel for the mechanism but is far outside the scope of our current project. Another piece of this project that would be interesting to add on to would be broadening the breadth of what language is being looked at. We considered poetry at first when figuring out what direction to move with the project and there are limitless examples of text that could be generated: poetry, movie scripts, books, dialogue, etc. There could be something compelling about seeing how different types and structures of language affect the output of our model.

Conclusion

Overall, this project presented a fun and interesting foray into the world of natural language processing and its application to song lyrics and music. While there are places for improvement and the door is open for future work to be appended, our model showed significant functionality in producing new song lyrics. Documentation can be found in the works cited section below and code can be found in the github repository linked below as well.

Works Cited

Greater Good: How Music Bonds Us Together
Psychology Today: Music and Language
Oxford Dictionary: Art Definition
MusicOSet: Enhanced Music Dataset
Project Inspiration from ActiveState: Lyrics Generator with Python
InferKit: Generate Text Demo
Medium: Attention in RNNs
Github Repository: CS4120 Final Project