

# Generating Doja Cat Song Lyrics

Emma Mavis

IDS 703 Fall 2021

[GitHub Repository](#)

*Note: The following content contains some explicit language*

## **I. Chosen NLP Problem: Language Modeling**

In this project, I have chosen to generate song lyrics from artist Doja Cat. Her song lyrics from three of her albums were grabbed from the lyric website *azlyrics*.

## **II. Generative Probabilistic Language Model**

A Markov Text generation algorithm was used as the probabilistic model for generating lyrics on a word-level basis. In this model, the input is a starting sentence (any chosen length of words), a corpus of text data, an integer for the length of the n-gram, and a deterministic flag. The output is a chunk of text (any chosen length of words) that represent lyrics where each word is determined by the most-probable next word calculation.

### **i. Performance Evaluation**

A word-level text generation has a large data set to search through, relative to a character level, so inherently will take longer to complete. With that said, with the small seed sentence and small number of iterations that were used in this project, the Markov Text generation model performed perfectly quickly. This model performs well when the length of the n-gram is around 8-12 and the model is asked to generate a short sequence of words. When the n-gram length is any lower or the requested number of words to generate is large, the output text becomes extremely repetitive. For example, with the seed, “u go to town go down,” and the length of requested text is 100 words,

n = 12 produces:

“u go to town go down go down go down yeah let me see you go to town yeah go down go down go down yeah yeah yeah if you 're down boy really down baby let me watch you go to town it 's your one chance baby never or now yeah let me see you go to town baby go to town yeah baby let me watch you go to town it 's your one chance baby never or now yeah let me see you go to town baby go to town yeah baby let me watch you”

n = 20 produces:

“u go to town go down go down go down yeah let me see you go to town yeah go down go down go down yeah let me see you go to town go down go down go down yeah let me see you go to town go down go down go down yeah let me see you go to town yeah go down go down go down yeah let me see you go to town go down go down go down yeah let me”

The text generated with the smaller  $n$  is much less repetitive, i.e., potentially more accurate.

## **III. Discriminative Neural Network Language Model**

The *Keras* library was used to implement a LSTM recurrent neural network model on a character-level basis. This model is an instance of *Sequential* and consists of one LSTM layer and one dense layer with a *softmax* activation. It is then compiled with a *categorical cross entropy* as the chosen loss in the optimizer. The corpus of data was chunked into training sequences of text and a sequence of next

characters. Input and output matrices are then created with label-encoded (one-hot encoded) character data so the model can be fit on the data with a designated batch size and number of epochs.

#### **i. Performance Evaluation**

This model is trickier to evaluate because I do not have much knowledge of machine learning methods and only learned the small amount of Keras needed to build and fit this model. That said, I believe the character-level generation has produced better text than the word-level generation. I was not able to run this model with very many epochs due to limits on my machine's processing power and memory, but I have produced results after three epochs. After each epoch iteration, the loss of the model decreases. For example, the loss after the first iteration was 0.8150, and after the second iteration the loss was 0.5502. The results have shown that as more epochs are run, the more the text begins to create actual words rather than nonsensical characters to put together. Additionally, when the diversity/temperature value is lower, the output becomes more sensible. For example, with the seed, "if she won't go dijo,"

Diversity = 0.5 produces:

"lean and your hands down woman woman woman ayy i can be your woman  
woman woman woman ayy i can get believe i got a man but i want a man to stand  
and fight stand and fight for meging i have it ain't got your ganester ass me ath no  
got a h baby i how ain 't coldow ha om ain't gotta ally want me a to you boy i 'll  
show you yeah baby let me watch you go to town it's your one chaine a manted get  
into"

Diversity = 1.2 produces:

"n 'cause yours cuap griss 'bout to fuckin' backa woo shised turn yeah reen to you  
vibpide let now douge imalesto make relall the sine orenan get whow thot woman  
nomy kild nice with my hain and side up misty i me and thought the erping  
everythith just a starde nah no girash and the sealy your anyther h all me him you  
pite night in a fastasi gato like a tiol ove your whole i won 't bed fow i 'm kid"

The lower diversity value produces text that spells words more accurately rather than the more nonsensical words produced by the higher diversity.

#### **IV. Pros and Cons of Each Method**

The quality of the probabilistic model is much better than that of the LSTM model, since lyrics are generated word by word rather than by characters. Of course, this method scaled up would take much more time through each iteration and potentially be less efficient. On a scaled-up context, the LSTM model would theoretically perform much better, but I could not produce those results because of hardware limitations.

Each model also required different formats of input data, which is tedious to change and/or automate. That said, the "correctness" of the LSTM model was higher than the probabilistic, which is expected because results are produced on a much more granular level.

In order to make the "best" machine learning model, additional tools such as GPU computation are necessary in order to run through enough epochs to generate more sensible text. Of course, interpretability of the lyrics generated are subject to context limitations – rap lyrics, especially those written by Doja Cat, are quite choppy and simple, with tons of vernacular. I believe it is tricky to interpret her lyrics in general, let alone machine-generated ones.

Moving forward, the following limitations would need to be addressed:

- Computation power
- Memory
- Data acquisition

- Machine learning (Keras) skills

#### References

<https://towardsdatascience.com/generating-drake-rap-lyrics-using-language-models-and-lstms-8725d71b1b12>

[https://keras.io/examples/generative/lstm\\_character\\_level\\_text\\_generation/](https://keras.io/examples/generative/lstm_character_level_text_generation/)

Wang, Patrick; Markov Text Generation Assignment; IDS 703 Fall 2021; September 8, 2021