RGPT-2: Rap Generator Based on GPT-2

Kurtis Shen
Columbian College of Arts & Sciences
George Washington University
Washington, DC 20052
lootlecoop@gwmail.gwu.edu

Hsueh-Yi Lu
Columbian College of Arts & Sciences
George Washington University
Washington, DC 20052
hsuehyi_lu2580@gwmail.gwu.edu

Abstract

This project is aimed to generate rap verses humanly realistic enough to be distinguishable from other genres. Our model – RGPT-2 – is based on fine-tuning the most prominent text generation model GPT-2, hence the name. Unlike traditional approaches, our implementation was heavily inspired by another sister model GPoet-2, utilizing a two-model consecutive generating process instead of only one. Uniqueness relies on the second model being trained on the reversed training texts, resulting in capturing more structure within a verse than training only the original texts. With no words being provided, our model was able to generate hyper-human like rap verses that could be easily categorized into hip-hop genre.

Table of Contents

1. Introduction

NLP is a developing technology with many issues and technical challenges that we need to overcome. With the advancement of NLP technology and new algorithms and structures, we began to overcome the previous difficulties step by step, and what we want to try in this project is to use the language generation model to train a rapper model, a model with the ability to create rap lyrics. Among many well-known models, the vast majority of models use as many fields as possible to train language models, so if we only input rap lyrics data, a model based on rap lyrics will be generated, and then use the classification model to identify the lyrics generated by the rap model, so as to judge whether these generated sentences can be judged as rap lyrics.

2. Related Work

2.1. Language Generation

Natural language generation (NLG) is a software process that produces natural language output. In one of the most widely-cited survey of NLG methods, NLG is characterized as "the subfield of artificial intelligence and computational linguistics that is concerned with the construction of computer systems than can produce understandable texts in English or other human languages from some underlying non-linguistic representation of information".

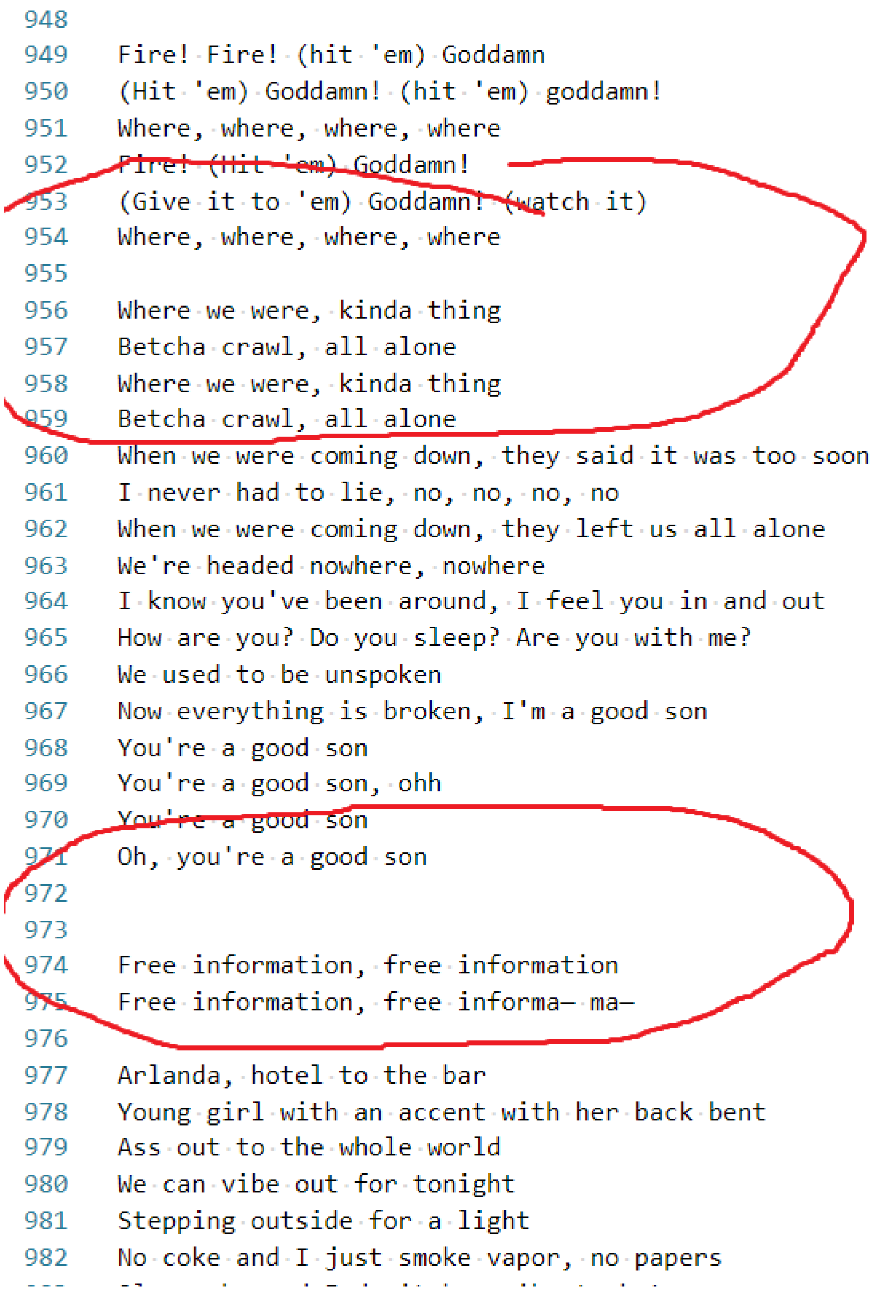
3. Datasets

3.1. Source

The dataset used in this project is from the Hip-Hop Encounters Data Science and Music Genre Classification from Kaggle. Hip-Hop Encounters Data Science contains name of several rappers and verses of their songs. Music Genre Classification contains a train dataset and test dataset, which both includes lyrics and its related music genre. There are two types of genres, rap and pop, and was classed by checking from Genius.com.

3.1.1. Text Generation

Training sets and validation sets are retrieved from a public open-source GitHub repository (fpaupier, 2018). They come in .txt format, and each file represents an artist. There are 39 artists in total. Each file consists of several songs by that particular artist. Songs are separated by two lines. Verses (paragraphs) are separated by one line. And each bar (one sentence) stands on their own line.



Fire! Fire! (hit 'em) Goddamn
(Hit 'em) Goddamn! (hit 'em) goddamn!
Where, where, where, where
~~Fire! (Hit 'em) Goddamn!~~
(Give it to 'em) Goddamn! (watch it)
Where, where, where, where

Where we were, kinda thing
Betcha crawl, all alone
Where we were, kinda thing
Betcha crawl, all alone
When we were coming down, they said it was too soon
I never had to lie, no, no, no, no
When we were coming down, they left us all alone
We're headed nowhere, nowhere
I know you've been around, I feel you in and out
How are you? Do you sleep? Are you with me?
We used to be unspoken
Now everything is broken, I'm a good son
You're a good son
You're a good son, ohh
~~You're a good son~~
Oh, you're a good son

Free information, free information
Free information, free informa— ma—

Arlanda, hotel to the bar
Young girl with an accent with her back bent
Ass out to the whole world
We can vibe out for tonight
Stepping outside for a light
No coke and I just smoke vapor, no papers

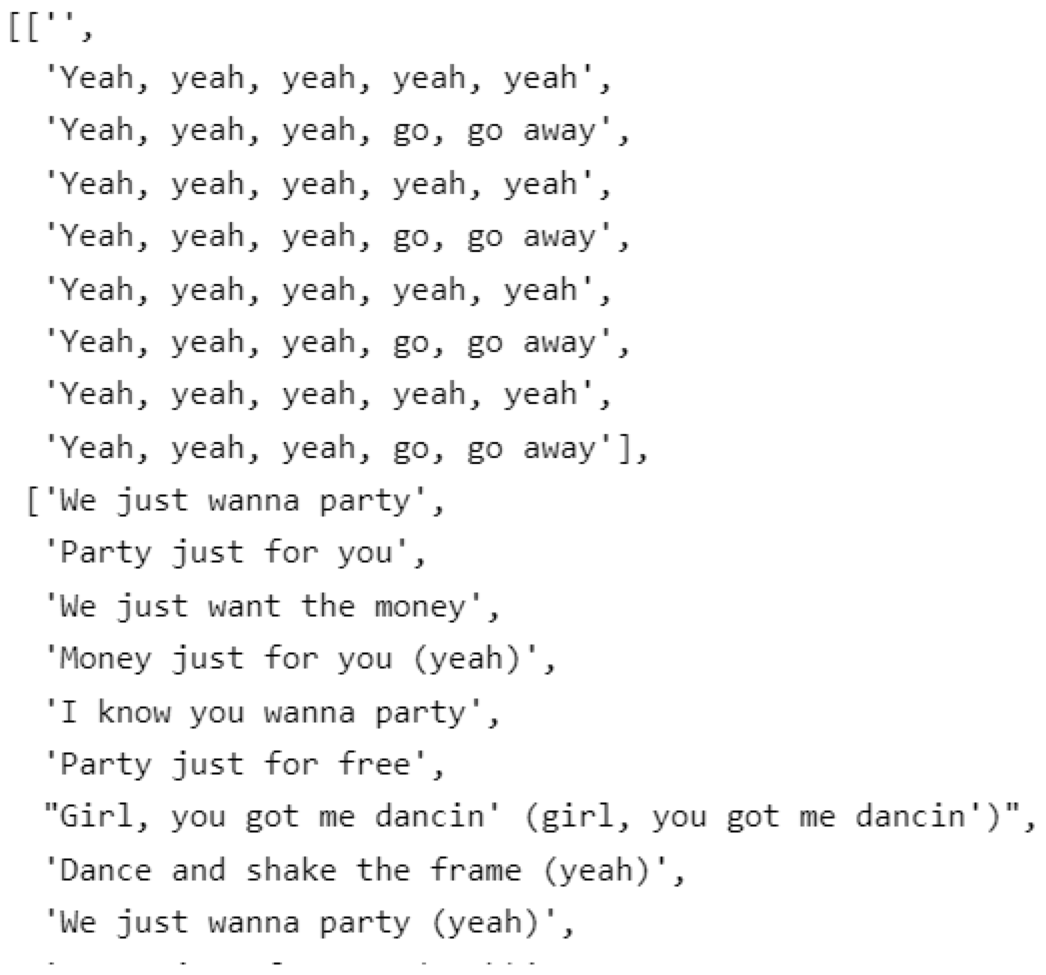
Figure 1 One Example Data File

3.1.2. Genre Classification

Training sets and testing sets are retrieved from a public music platform Genius.com. They come in .csv format, and both are in the same format. Each file contains string of lyrics and its related music genre, pop song or hip-hop music. 2584 lyric samples for the testing set are with ids and 35726 samples for the training set.

3.2. Preprocessing

Based on our file template, our final dataset should be list of lists, with each list representing a verse. A three step split process to split the texts, from splitting songs from two lines, to splitting verses within the song, to splitting bars within the verses, was deployed, with the final split doing double list comprehension to avoid creating an extra nested list.



```
[['',
  'Yeah, yeah, yeah, yeah, yeah',
  'Yeah, yeah, yeah, go, go away',
  'Yeah, yeah, yeah, yeah, yeah',
  'Yeah, yeah, yeah, go, go away',
  'Yeah, yeah, yeah, yeah, yeah',
  'Yeah, yeah, yeah, go, go away',
  'Yeah, yeah, yeah, yeah, yeah',
  'Yeah, yeah, yeah, go, go away'],
 ['We just wanna party',
  'Party just for you',
  'We just want the money',
  'Money just for you (yeah)',
  'I know you wanna party',
  'Party just for free',
  "Girl, you got me dancin' (girl, you got me dancin')",
  'Dance and shake the frame (yeah)',
  'We just wanna party (yeah)',
```

Figure 2 Post_processed Datassets

Since we are doing unconditional text generation (text generation without providing any contexts or words), we do not need a test dataset. Thus, all these verses were divided into training and just validation set.

4. Models

4.1. Inspiration

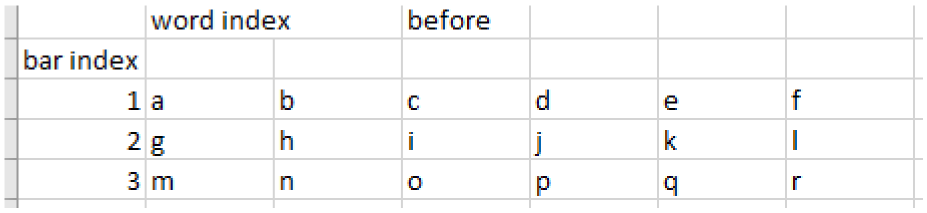
Our model is based on GPT-2. It was created in 2019 by OpenAI **(OpenAI, 2019)**. It can perform a wide range of natural language tasks, like translation, summarization, generation, etc., just like its predecessor GPT, but is a scale-up version (Radford, Narasimhan, Salimans, & Sutskeve). With our intensions to generate unconditioned texts, GPT-2 seemed an adequate fit, supported by the wide usage among various fields of researchers and practitioners. However, based on research, albeit impressive results of occasional indistinguishable human-like turnouts (Kaiser, 2020), it tends to yield repetitive or non-sensical messages when generating lengthy sentences **(Vincent, 2019)**. Despite these concerns, related work such as GPoet-2 producing more convincing results, capturing more structure **(Lo, Ariss, & Kurz, 2022)**, encouraged us to customize GPT-2 and create our own variant. Contrary to almost all approaches working on restructuring model architectures, we were exceptionally elevated and inspired by the great work from architects behind GPoet-2 to approach from another angle. As has been taught since the first day we started studying the field, data is the key. Indeed, models share medals for success, feature engineering is and forever will be the most important step.

The problem with poor results from generating large texts using GPT-2 lies from the structure capturing. With attention and mask, our inputs will have sights of succeeding words and sentences. This is how the generated texts have subject continuity. In our specific task, however, subject continuity is the basic requirement. Rap song separates itself from other block of texts due to rhyming words between sentences, and a lot of the time, rhymed words appeared at the same position within a bar (sentence). The task inspired the method utilized behind GPoet-2 was to generate limericks **(Lo, Ariss, & Kurz, 2022)**. They are five-line humorous, sometimes rude, poems following AABBA structure. Authors came up with the idea of training the text in reverse in each sentence after normal training. Although intuition seems unclear at first sight, humans' greatest ability is imitation. With this ability, we were able to imitate birds to create planes, imitate fish to create boats, conquering the impossible. To let the computer behave more like humans, we have to exploit the techniques that made us human beings great at what we do in the first place.

When rapping songs, after one bar, in order to rhyme with the previous bar, we continue the subject while constantly referring to the previous bar, sometimes even further, to subconsciously notify us how many words can we still fit in the bar, and where our rhyming words should go. Yes, sequential texts make sure subject is continued, but without constantly looking back, the further away from the first bar, the further away the subject is escaping.

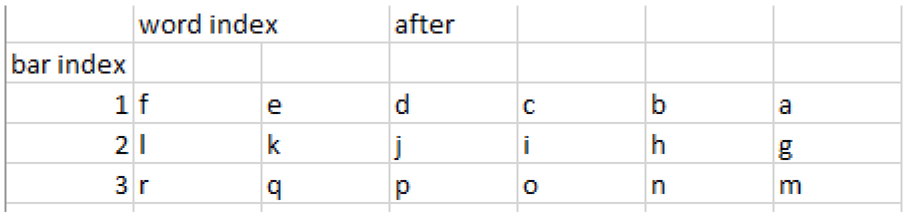
4.2. Framework

To implement this idea. We trained the standard GPT-2 model on our lyrics set. Then we trained the standard GPT-2 model again on the reversed dataset, reversed being each sentence was reversed with the last word now being the first, while sentence position within the verse (paragraph) remains the same. To better visualize, graphs of before and after transforming is shown below.



	word index		before			
bar index						
1	a	b	c	d	e	f
2	g	h	i	j	k	l
3	m	n	o	p	q	r

Figure 3 Before Reversing



	word index		after			
bar index						
1	f	e	d	c	b	a
2	l	k	j	i	h	g
3	r	q	p	o	n	m

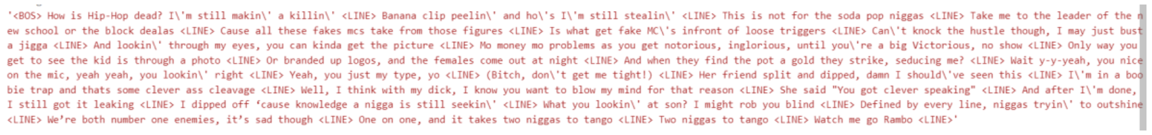
Figure 4 After reversing

After training, unconditional texts were generated using the model trained on standard-order texts. However, during the generation phase, only the first line was generated by the standard model, since it specializes at producing more diversified topics than the reversed one. Then subsequent lines were generated inputting the first line using the reverse-order model.

4.2.1. Training

For better capturing the structure, we add three special tokens to our tokenizer: verse beginning token (paragraph starts), padding token and bar ending token (sentence ends). Following standard procedure, we subclass dataset class for pipelining data retrieval. We used customized class batch collation method to retrieve batches, encapsulating sequential text and reversed text tokenization method from the script.

Reversing the verse is done by creating an identical sized list of input ids. Then iterate from the index after beginning token till the end of sentence token. Reverse the bar's tokens and store them into the identical list. And next, move onto the next bar (sentence).



```
'<BOS> How is Hip-Hop dead? I\'m still makin\' a killin\' <LINE> Banana clip peelin\' and ho\'s I\'m still stealin\' <LINE> This is not for the soda pop niggas <LINE> Take me to the leader of the n
ew school or the block dealas <LINE> Cause all these fakes mcs take from those figures <LINE> Is what get fake MC\'s infront of loose triggers <LINE> Can\'t knock the hustle though, I may just bust
a jigga <LINE> And lookin\' through my eyes, you can kinda get the picture <LINE> Mo money mo problems as you get notorious, inglorious, until you\'re a big Victorious, no show <LINE> Only way you
get to see the kid is through a photo <LINE> Or branded up logos, and the females come out at night <LINE> And when they find the pot a gold they strike, seducing me? <LINE> Wait y-y-yeah, you nice
on the mic, yeah yeah, you lookin\' right <LINE> Yeah, you just my type, yo <LINE> (Bitch, don\'t get me tight!) <LINE> Her friend split and dipped, damn I should\'ve seen this <LINE> I\'m in a boo
bie trap and thats some clever ass cleavage <LINE> Well, I think with my dick, I know you want to blow my mind for that reason <LINE> She said "You got clever speaking" <LINE> And after I\'m done,
I still got it leaking <LINE> I dipped off ‘cause knowledge a nigga is still seekin\' <LINE> What you lookin\' at son? I might rob you blind <LINE> Defined by every line, niggas tryin\' to outshine
<LINE> We’re both number one enemies, it’s sad though <LINE> One on one, and it takes two niggas to tango <LINE> Two niggas to tango <LINE> Watch me go Rambo <LINE>'
```

Figure 5 Example Verse

Model is the GPT2head model with resized token embeddings to account for adding special tokens.

With optimizer, learning rate scheduler and setting up, our model began training to minimize the negative log-likelihood loss.

4.2.1.1. Metrics

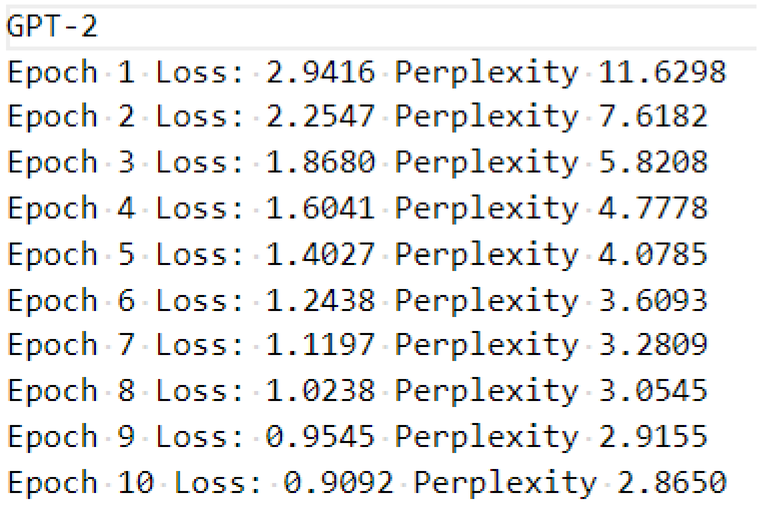
The metric to compare performance of each epoch in order to save the best trained parameters in this case is perplexity. Perplexity in the natural language domain is based on perplexity of a probability model.

$$PP(p) = e^{H(p)}$$

$$\text{where } H(\tilde{p},\ q) = -\sum_x \tilde{p}(x) \log_2 q(x)$$

$$\text{where } \tilde{p}(x) = \frac{n}{N}$$

Perplexity is used to approximate the unknown probability distribution p of a model. The idea is drawing a training sample from p. Preparing a proposed probability model q, perplexity evaluate q by predicting another sample from p. $H(\tilde{p},\ q)$ here is the cross-entropy, where $\tilde{p}(x)$ represents the factor of the amount of times x appears in the sample of size N n. In the context of natural language model, by custom, we normalize the number by sentence length **(Wikipedia, n.d.)**.

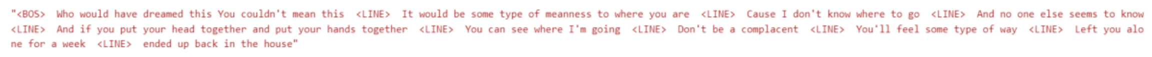


```
GPT-2
Epoch 1 Loss: 2.9416 Perplexity 11.6298
Epoch 2 Loss: 2.2547 Perplexity 7.6182
Epoch 3 Loss: 1.8680 Perplexity 5.8208
Epoch 4 Loss: 1.6041 Perplexity 4.7778
Epoch 5 Loss: 1.4027 Perplexity 4.0785
Epoch 6 Loss: 1.2438 Perplexity 3.6093
Epoch 7 Loss: 1.1197 Perplexity 3.2809
Epoch 8 Loss: 1.0238 Perplexity 3.0545
Epoch 9 Loss: 0.9545 Perplexity 2.9155
Epoch 10 Loss: 0.9092 Perplexity 2.8650
```

Figure 6 Standard Model Validation Loss and perplexity

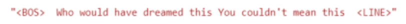
4.2.2. Generating

After training, our model is capable of doing unconditional text generating. The method for generating is a two-step method. First, we use the standard trained model to generate the first line, having explained that the standard model has the capability of generating a variety of topics. Because of the use of end of line token, it is extremely easy to slice the first bar (sentence). Simply split on the <LINE> string and pick the first element.



```
"<BOS>  Who would have dreamed this You couldn't mean this  <LINE>  It would be some type of meanness to where you are  <LINE>  Cause I don't know where to go  <LINE>  And no one else seems to know
<LINE>  And if you put your head together and put your hands together  <LINE>  You can see where I'm going  <LINE>  Don't be a complacent  <LINE>  You'll feel some type of way  <LINE>  Left you alo
ne for a week  <LINE>  ended up back in the house"
```

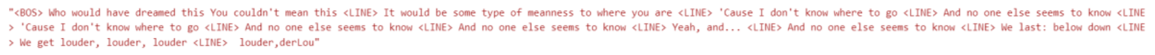
Figure 7 Standard Model Generating The First Verse



```
"<BOS>  Who would have dreamed this You couldn't mean this  <LINE>"
```

Figure 8 Using only the first bar

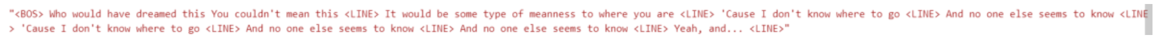
Next step is to feed the first line as out input to generate the remaining bars using the reverse trained model. Because our second model was trained on reverse-ordered bar (sentence), the first sentence will have to be reversed. However, our final output needs to be in the correct order, when decoding the batch, we also need to again reverse the input ids back to normal order.



```
"<BOS> Who would have dreamed this You couldn't mean this <LINE> It would be some type of meanness to where you are <LINE> 'Cause I don't know where to go <LINE> And no one else seems to know <LINE
> 'Cause I don't know where to go <LINE> And no one else seems to know <LINE> And no one else seems to know <LINE> Yeah, and... <LINE> And no one else seems to know <LINE> We last: below down <LINE
> We get louder, louder, louder <LINE>  louder,derLou"
```

Figure 9 Generated Verse

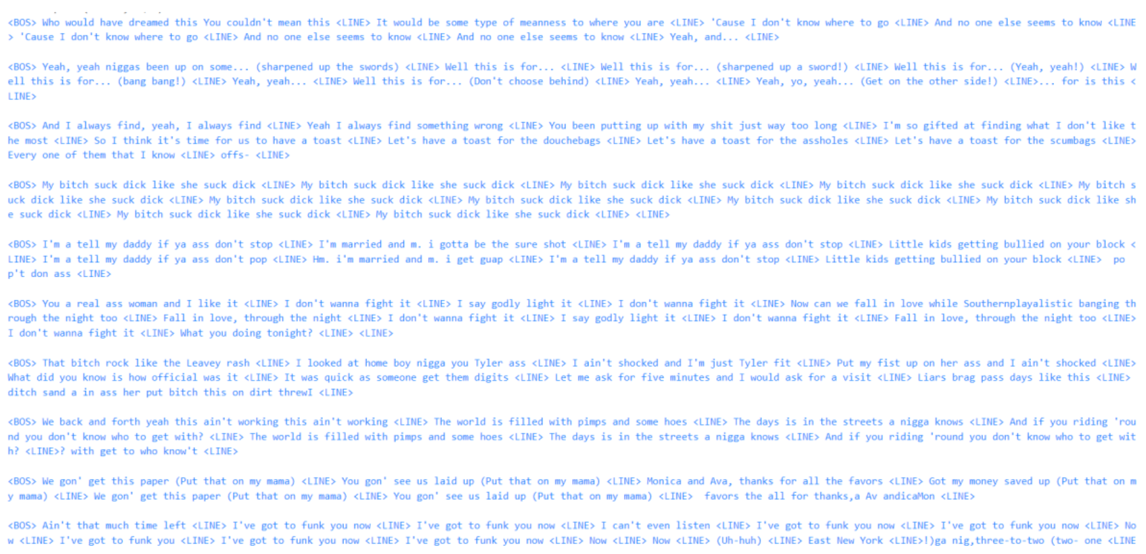
Normally, a verse has either 8 or 16 bars. Thus, we randomly chose to slice either 8 or 16 lines.



```
"<BOS> Who would have dreamed this You couldn't mean this <LINE> It would be some type of meanness to where you are <LINE> 'Cause I don't know where to go <LINE> And no one else seems to know <LINE
> 'Cause I don't know where to go <LINE> And no one else seems to know <LINE> And no one else seems to know <LINE> Yeah, and... <LINE>"
```

Figure 10 Processed Generated Verse

And finally, we generated a lengthy “song” with arbitrary numbers of verses.



```
<BOS> Who would have dreamed this You couldn't mean this <LINE> It would be some type of meanness to where you are <LINE> 'Cause I don't know where to go <LINE> And no one else seems to know <LINE
> 'Cause I don't know where to go <LINE> And no one else seems to know <LINE> And no one else seems to know <LINE> Yeah, and... <LINE>

<BOS> Yeah, yeah niggas been up on some... (sharpened up the swords) <LINE> Well this is for... <LINE> Well this is for... (sharpened up a sword!) <LINE> Well this is for... (Yeah, yeah!) <LINE> W
ell this is for... (bang bang!) <LINE> Yeah, yeah... <LINE> Well this is for... (Don't choose behind) <LINE> Yeah, yeah... <LINE> Yeah, yo, yeah... (Get on the other side!) <LINE>... for is this <
LINE>

<BOS> And I always find, yeah, I always find <LINE> Yeah I always find something wrong <LINE> You been putting up with my shit just way too long <LINE> I'm so gifted at finding what I don't like t
he most <LINE> So I think it's time for us to have a toast <LINE> Let's have a toast for the douchebags <LINE> Let's have a toast for the assholes <LINE> Let's have a toast for the scumbags <LINE>
Every one of them that I know <LINE> offs- <LINE>

<BOS> My bitch suck dick like she suck dick <LINE> My bitch suck dick like she suck dick <LINE> My bitch suck dick like she suck dick <LINE> My bitch suck dick like she suck dick <LINE> My bitch s
uck dick like she suck dick <LINE> My bitch suck dick like she suck dick <LINE> My bitch suck dick like she suck dick <LINE> My bitch suck dick like she suck dick <LINE> My bitch suck dick like sh
e suck dick <LINE> My bitch suck dick like she suck dick <LINE> My bitch suck dick like she suck dick <LINE> <LINE>

<BOS> I'm a tell my daddy if ya ass don't stop <LINE> I'm married and m. i gotta be the sure shot <LINE> I'm a tell my daddy if ya ass don't stop <LINE> Little kids getting bullied on your block <
LINE> I'm a tell my daddy if ya ass don't pop <LINE> Hm. i'm married and m. i get guap <LINE> I'm a tell my daddy if ya ass don't stop <LINE> Little kids getting bullied on your block <LINE>  po
p't don ass <LINE>

<BOS> You a real ass woman and I like it <LINE> I don't wanna fight it <LINE> I say godly light it <LINE> I don't wanna fight it <LINE> Now can we fall in love while Southernplayalistic banging th
rough the night too <LINE> Fall in love, through the night <LINE> I don't wanna fight it <LINE> I say godly light it <LINE> I don't wanna fight it <LINE> Fall in love, through the night too <LINE>
I don't wanna fight it <LINE> What you doing tonight? <LINE> <LINE>

<BOS> That bitch rock like the Leavey rash <LINE> I looked at home boy nigga you Tyler ass <LINE> I ain't shocked and I'm just Tyler fit <LINE> Put my fist up on her ass and I ain't shocked <LINE>
What did you know is how official was it <LINE> It was quick as someone get them digits <LINE> Let me ask for five minutes and I would ask for a visit <LINE> Liars brag pass days like this <LINE>
ditch sand a in ass her put bitch this on dirt threwI <LINE>

<BOS> We back and forth yeah this ain't working this ain't working <LINE> The world is filled with pimps and some hoes <LINE> The days is in the streets a nigga knows <LINE> And if you riding 'rou
nd you don't know who to get with? <LINE> The world is filled with pimps and some hoes <LINE> The days is in the streets a nigga knows <LINE> And if you riding 'round you don't know who to get wit
h? <LINE>? with get to who know't <LINE>

<BOS> We gon' get this paper (Put that on my mama) <LINE> You gon' see us laid up (Put that on my mama) <LINE> Monica and Ava, thanks for all the favors <LINE> Got my money saved up (Put that on m
y mama) <LINE> We gon' get this paper (Put that on my mama) <LINE> You gon' see us laid up (Put that on my mama) <LINE>  favors the all for thanks,a Av andicaMon <LINE>

<BOS> Ain't that much time left <LINE> I've got to funk you now <LINE> I've got to funk you now <LINE> I can't even listen <LINE> I've got to funk you now <LINE> I've got to funk you now <LINE> No
w <LINE> I've got to funk you <LINE> I've got to funk you now <LINE> I've got to funk you now <LINE> Now <LINE> Now <LINE> (Uh-huh) <LINE> East New York <LINE>!)ga nig,three-to-two (two- one <LINE
```

Figure 11 Verses Snippet

4.2.3. Classification Model for Generator Evaluation

For the classification model, we are using a combined model which includes a BERT model as the base model and a classifier model for the final output.

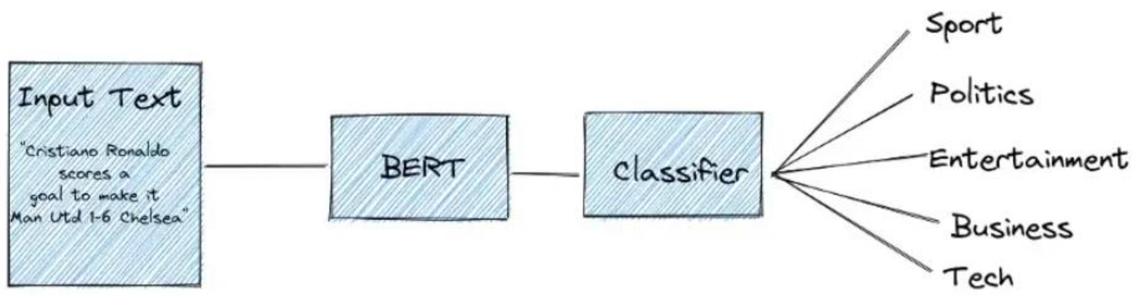

Input Text
"Cristiano Ronaldo scores a goal to make it Man Utd 1-6 Chelsea"
BERT
Classifier
Sport
Politics
Entertainment
Business
Tech

Figure 12 Model structure

BERT is a transformer-based machine learning technique for natural language processing (NLP) pre-training developed by Google. The two major benefits of using BERT as a base model is that first, BERT is a pre-trained on unlabeled data extracted from BooksCorups, which has 800M words, and from Wikipedia, which has 2,500M words. Second point is that as the name suggests, it is pre-trained by utilizing the bidirectional nature of the encoder stacks. This means that BERT learns information from sequence of words not only left to right, but also from right to left.

After translating the data using CustomDataset, we collected the label and the embedded tokens (sentences) as the input of the combined BERT base model. Now this model can classify the input lyric sentence is more like a pop music or a rap music.

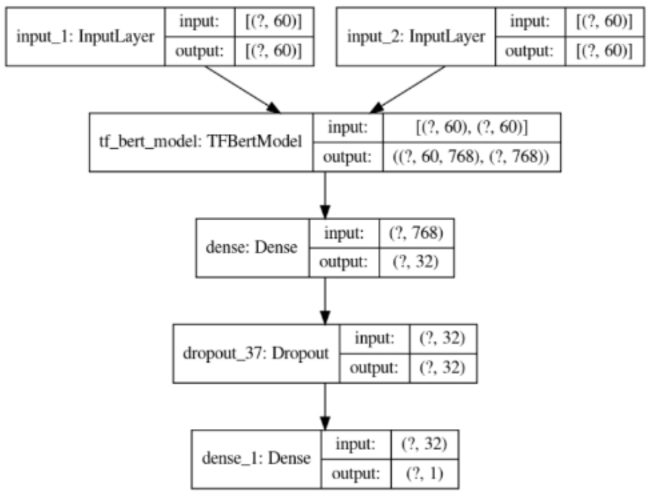


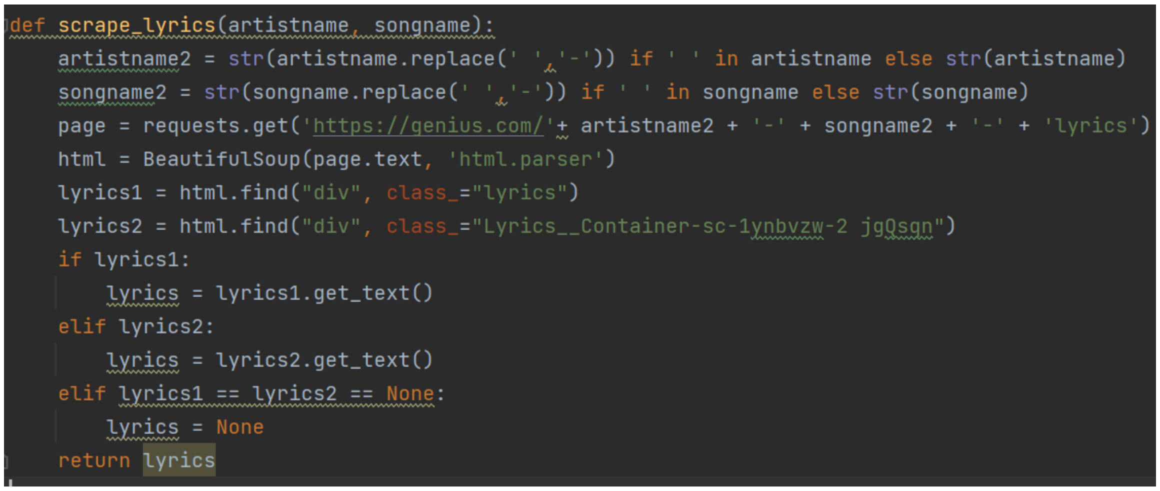
Figure 13 Model Structure Example Figure

5. Results

For the results, we pass several lyrics generated by the GPT-2 model to the pop or rap music classification model to check how the classification model classify the generated lyrics. After testing, 87 percent of the lyrics were considered to be lyrics of rap music lyrics. Because of the BERT model is trained based on data from different fields, this model can be regarded as a judgment model that is not biased and has cognition in various fields, so it won't be affected by specific topics. Due to that most of the lyrics of most raps are related to specific issues (such as gangsters, drugs, violence) or specific words (the F word or the N word), the language generation model trained based on such data will have serious generation bias, and that is what we hope of. Since when we use two models trained by different data as the judgment benchmark, it will not be affected by overfitting problem.

6. Future Work

Overall, the project was considered a success based on the results. The final rap lyrics discriminative ratios met our initial expectations, although our report could have a lot of space for improvement. For example, in terms of database creation and data acquisition, we tried to design a program that uses beautiful soup to automatically download a large amount of lyrics data from music platforms. Increasing the amount of data is a very direct way to improve the model. We can automatically download from various websites to further improve the quality of the two models, especially in the classification model. If you can obtain more different types of lyrics and effectively classify them and add them to the database, the entire classification model will be more reasonable and completed.



```
def scrape_lyrics(artistname, songname):
    artistname2 = str(artistname.replace(' ','-')) if ' ' in artistname else str(artistname)
    songname2 = str(songname.replace(' ','-')) if ' ' in songname else str(songname)
    page = requests.get('https://genius.com/'+ artistname2 + '-' + songname2 + '-' + 'lyrics')
    html = BeautifulSoup(page.text, 'html.parser')
    lyrics1 = html.find("div", class_="lyrics")
    lyrics2 = html.find("div", class_="Lyrics__Container-sc-1ynbvzw-2 jgQsgn")
    if lyrics1:
        lyrics = lyrics1.get_text()
    elif lyrics2:
        lyrics = lyrics2.get_text()
    elif lyrics1 == lyrics2 == None:
        lyrics = None
    return lyrics
```

Figure 14 Lyrics Downloading Code which is not completed

The other point lies in the design of indicators. In this project, we only used the classification model of pop music and rap music to test our results. Rap music can be classified by more indicators, such as the number of rhymes and the composition of sentences. The way and the classification of most words or the number of repetitions of lyrics. How to quantify these values and design them into mathematical formulas that can be accepted by the model is what we should consider.

$$D_{\text{rhyme}} = \frac{R(l_1, l_2) + R(l_3, l_4) + R(l_1, l_5) + R(l_2, l_5)}{4}$$

Figure 15 Limerick Rhyming Structure Performance Equation

7. Conclusion

Thinking from the data perspective instead of models makes our customized RGPT-2 model being able to generate human-like verses, while maintaining the rap structure of a verse.

AI is drastically changing our world. Like the fresh chatGPT being able to engage in even philosophical conversation, maybe one day, we will be listening to AI rapping their own generated songs on Spotify

8. References

fpaupier. (2018, 8 11). *RapLyrics-Scraper.* Retrieved from Github: https://github.com/fpaupier/RapLyrics-Scraper/tree/master/lyrics_US

Kaiser, C. (2020, 1 31). *Too big to deploy: How GPT-2 is breaking servers*. Retrieved from towardsdatascience:

https://web.archive.org/web/20200215145640/https://towardsdatascience.com/too-big-to-deploy-how-gpt-2-is-breaking-production-63ab29f0897c

Lo, K.-L., Ariss, R., & Kurz, P. (2022, 5 18). *GPoeT-2: A GPT-2 Based Poem Generator.* Retrieved from arxiv: https://arxiv.org/abs/2205.08847

OpenAI. (2019, 2 14). *Better Language Models.* Retrieved from OpenAI: https://openai.com/blog/better-language-models/

Radford, A., Narasimhan, K., Salimans, T., & Sutskeve, I. (n.d.). *Improving Language Understanding.* Retrieved from OpenAI: https://cdn.openai.com/research-covers/language-unsupervised/language_understanding_paper.pdf

Vincent, J. (2019, 11 7). *OpenAI has published the text-generating AI it said was too dangerous to share*. Retrieved from theverge: https://www.theverge.com/2019/11/7/20953040/openai-text-generation-ai-gpt-2-full-model-release-1-5b-parameters

Wikipedia. (n.d.). *Perplexity of a probability model*. Retrieved from Wikipedia: https://en.wikipedia.org/wiki/Perplexity

Wikipedia. (n.d.). *Natrual Language Generator*. Retrieved from Wikipedia:

https://en.wikipedia.org/wiki/Natural_language_generation

Text Classification with BERT in PyTorch. Retrieved from:

https://towardsdatascience.com/text-classification-with-bert-in-pytorch-887965e5820f