STATS 402 - Interdisciplinary Data Analysis
Stylistic Hip-hop Lyrics Generator Based on Transformer Model

Boqing Zheng, Shuhe Wang
bz88@duke.edu, sw505@duke.edu

Abstract

Hip-hop music, as a highly influential music style in modern days, has captivated global audiences with its unique linguistic creativity and storytelling prowess. The composition of hip-hop lyrics separates from other music forms by especially focusing on the integration of rhythm, rhyme, and wordplay. Plenty of works have already applied machine learning and deep learning in automatically generating hip-hop lyrics, while some of the lyrics they generated still lack fluency, continuity, or distinctive style. Our project is aiming to design an automatic hop-hop lyrics generator using a transformer-based language model that can balance the rhyming and reasonability of the lyrics. By learning the algorithm between the words and rhymes from the lyrics of a specific rapper, the model should generate several lines of lyrics following a given prompt within the rapper's style. Our approach is to use datasets scraped from a specific hip-hop lyrics website to train our language model, for which the datasets could be lyrics written by several famous rappers to limit the generated lyrics to specific styles. Our model was built based on an existing GPT-2 model with improvements and modifications. We combined two methods to improve the quality and effectiveness of our generated lyrics, one is using functions after the generating process to make the endings of each line rhyme, the other is to modify the loss function to consider the rhyme density into the training process. As a result, our model is able to generate lyrics with high rhyme density, continuity, and similarity with the style of the rapper we chose. The performance of the model was evaluated in both objective and subjective ways. The rhyme density of the generated lyrics and the perplexity of the trained language model were used to evaluate the validity of our results objectively. Surveys were conducted to let humans evaluate the quality of our generated lyrics subjectively. Furthermore, the comparison between our generated lyrics and the original rappers' lyrics can show the improvements and limitations of our model.

1. Introduction

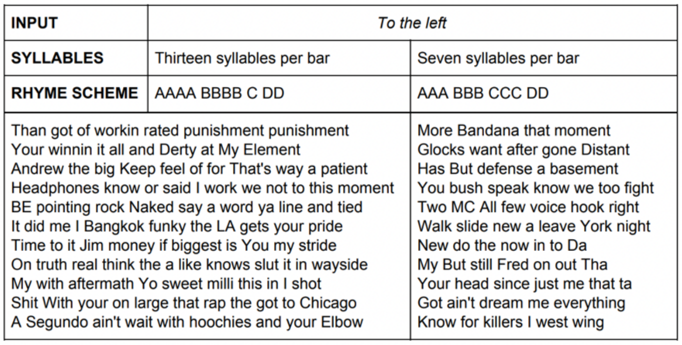
Since the 1970s, hip-hop music has been steadily gaining popularity among global youth [1]. Hip-hop artists often express their life perspectives and experiences through their songs, creating a strong impression and resonance among many listeners. The lyrics, rather than intricate melodies, have emerged as a fundamental component in crafting a rap song and conveying messages [2]. Most of the famous rappers have already established their own style of writing lyrics since several consistent aspects are required when composing hip-hop lyrics, such as rhyme, wordplay, flow, coherence, emotion, and depth of meaning [2]. Meanwhile, machine learning and deep learning methods came of age in recent years, which allow models to learn the algorithm of existing data and generate new ones by themselves [3] [4]. Since hip-hop lyrics have its common rules and styles, automatically generating hip-hop lyrics using machine learning technology has become an interesting field for researchers [5]. Plenty of approaches were applied to design a computational model which can generate hip-hop lyrics with continued meanings, while most of the results of previous works still lack fluency and quality. It is still a challenging task to let the model sufficiently and efficiently learn the rhyme algorithm and the common composing style of hip-hop lyrics at the same time. In addition, there is little previous study that generates hip-pop lyrics within a specific rapper's style. Therefore, we are aiming to design a language model which can generate high quality hip-hop lyrics with a given style. Improvements and comparison were also conducted for the existing models to examine the effectiveness of the current results.

After reviewing the previous models of generating rap lyrics, we designed our model based on the Transformer language model to learn, process, and generate our lyrics. Compared to other natural language processing models, the Transformer is popular in recent years for lyric generation and can perform better in learning the global content of the given text [6]. To make the generated lyrics rhyme and continue, we tried two ways that were commonly used in previous studies: one is to add filters after the lyrics generation to make the endings of each line rhyme, the other is to let the models learn the rhyme characteristics and directly generate rhymed lines [7]. By using lyrics from specific rappers to train our model, we expect our generated lyrics can imitate the style of the original rappers. Since the quality of rap lyrics connects to people's taste and emotion, we evaluated our model in both objective and subjective ways. Objectively, we used rhyme density to evaluate the generated lyrics and calculate perplexity to test the language model. Subjectively, surveys were conducted to let humans score the lyrics in several aspects. In the following sections, we will introduce the previous work as the background and explain our research process, technical routes, and evaluation methods in detail. Finally, we will analyze the performance of our model with valuable findings and potential improvements.

2. Related work

Prior research has successfully employed machine learning to understand the structure of rap lyrics and subsequently generate new ones. Malmi et al. used the RankSVM algorithm and a deep neural network model to develop a predictive model named *DeepBeat*, capable of composing succeeding lines to existing lyrics [8]. Later, Hernandez et al. utilized Word2Vec to "determine the appropriate window size, the word embeddings, and produce the verses", thereby enhancing the grammatical accuracy and fluency of the generated lyrics [5]. More recently, Xue et al. designed a Transformer-based system to generate rap lyrics, constructing a model known as *DeepRapper* that can even generate both rhymes and rhythm [7]. For our project, we chose to create a model based on the Transformer for natural language processing (NLP) and related word association. The Transformer model is capable of assigning weight to each word in a sequence based on its significance relative to other words, enabling it to better capture long-distance dependencies and context compared to other techniques [6]. In contrast to a convolution neural network (CNN), the Transformer is superior in considering the global context [9]. Maurício et al. also stated that a transformer-based architecture can perform better accuracy and efficiency in NLP tasks [10].

In current methodologies, the primary challenge lies in striking a balance between maintaining rhyme precision and ensuring the logic or quality of the lyrics. Previous systems for generating hip-hop lyrics either incorporate a unique token at the end-lines of a rap lyric to promote the learning [11] or employ a "two-stage strategy" for rhyme modeling that initially produces rap lyrics and subsequently appends rhyme tokens following the generated lyrics [12]. However, previous results show that the lyrics generated using the above methods can still lack fluency and effectiveness. Besides, the models cannot completely learn the rhyme algorithm and the added rhymed endings lack diversity. For example, as shown in Figure 1, the Word2Vec method seems only generate rhymed lyrics while losing the reasonability of the lyrics [5]. Xue et al. and Malmi et al. have tried to design new training processes to let the model learn and generate rhymed lyrics, while the style and structure of the output still seemed monotonous and simple [7] [8]. Therefore, our motivation is to find ways to improve the existing models to increase the creativity, rhyme diversity, and reasonability of the generated rap lyrics, and also try to generate stylistic rap lyrics of some specific famous rappers.



<table>
<tr><td><b>INPUT</b></td><td colspan="2"><i>To the left</i></td></tr>
<tr><td><b>SYLLABLES</b></td><td>Thirteen syllables per bar</td><td>Seven syllables per bar</td></tr>
<tr><td><b>RHYME SCHEME</b></td><td>AAAA BBBB C DD</td><td>AAA BBB CCC DD</td></tr>
<tr><td colspan="2">Than got of workin rated punishment punishment
Your winnin it all and Derty at My Element
Andrew the big Keep feel of for That's way a patient
Headphones know or said I work we not to this moment
BE pointing rock Naked say a word ya line and tied
It did me I Bangkok funky the LA gets your pride
Time to it Jim money if biggest is You my stride
On truth real think the a like knows slut it in wayside
My with aftermath Yo sweet milli this in I shot
Shit With your on large that rap the got to Chicago
A Segundo ain't wait with hoochies and your Elbow</td><td>More Bandana that moment
Glocks want after gone Distant
Has But defense a basement
You bush speak know we too fight
Two MC All few voice hook right
Walk slide new a leave York night
New do the now in to Da
My But still Fred on out Tha
Your head since just me that ta
Got ain't dream me everything
Know for killers I west wing</td></tr>
</table>

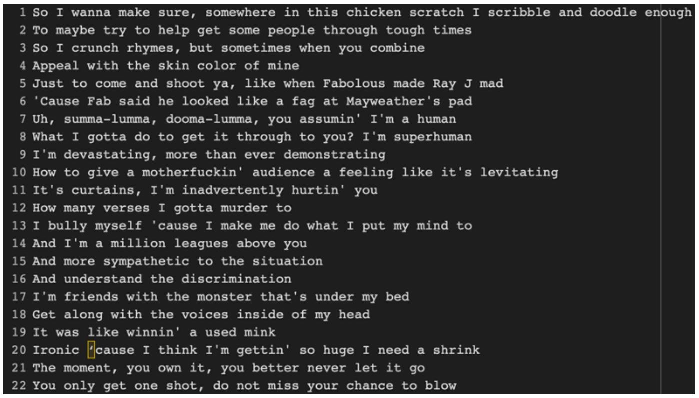
Figure 1. The sample lyrics from Hernandez et al. [5]

3. The proposed method

3.1 Data Acquisition and Data Cleaning

In our quest to gather rap lyrics from specific artists, we chose to extract data from Genius.com [13]. This platform offers a more extensive lyric database compared to other open-source hip-hop lyric resources and has diversified to include a wider range of content. We utilized the BeautifulSoup package to extract lyrics from Genius.com [14]. Once we obtained the client access token from Genius.com, we sent requests to the Genius API to gain access to the artists and their lyrics, which provided a json object in return [15]. Our code identifies songs on Genius.com by the names of specific rappers and retrieves the lyrics of those songs. We can specify the number of songs we want to extract for each rapper and store them in text files. The songs we obtain are listed in order of decreasing popularity. Currently, we scraped lyrics of three US rappers from the website using our code: Drake, A$AP Rocky, and Eminem. We extracted hundreds of songs for each rapper, yielding between three thousand and six thousand lines.

For the data cleaning process, we eliminated all "[Verse]" and "[Bridge]" labels and the spaces between verses. We also performed basic string cleaning on the lyrics by removing punctuation marks. These actions can make it easier for our model to learn the connections between words with similar or continued meanings. During later experiments, we noticed that our model consistently generated identical lines because the same lines can satisfy the requirements of rhyme and continuity. Thus, we went back to remove all the duplicated lines in the original datasets. Moreover, to improve the model's ability to learn the rhyme algorithm, we attempted to train our model solely with rhymed lines. Part of the dataset after data processing is shown in Figure 2.



```
So I wanna make sure, somewhere in this chicken scratch I scribble and doodle enough
To maybe try to help get some people through tough times
So I crunch rhymes, but sometimes when you combine
Appeal with the skin color of mine
Just to come and shoot ya, like when Fabolous made Ray J mad
'Cause Fab said he looked like a fag at Mayweather's pad
Uh, summa-lumma, dooma-lumma, you assumin' I'm a human
What I gotta do to get it through to you? I'm superhuman
I'm devastating, more than ever demonstrating
How to give a motherfuckin' audience a feeling like it's levitating
It's curtains, I'm inadvertently hurtin' you
How many verses I gotta murder to
I bully myself 'cause I make me do what I put my mind to
And I'm a million leagues above you
And more sympathetic to the situation
And understand the discrimination
I'm friends with the monster that's under my bed
Get along with the voices inside of my head
It was like winnin' a used mink
Ironic 'cause I think I'm gettin' so huge I need a shrink
The moment, you own it, you better never let it go
You only get one shot, do not miss your chance to blow
```

Figure 2. Part of the cleaned dataset

3.2 Transformer Based Model

Our overall model was eventually trained with GPT-2 (Generative Pretrained Transformer 2) as our model backbone. Transformers is a deep learning model structure based on the Self-Attention Mechanism, which has shown excellent performance in many tasks in the field of NLP (natural language processing) [6]. The main advantage of Transformers is that it is better able to understand the global context than other deep learning models [16]. This gives the Transformer class model the advantage of continuity in lyrics creation. GPT-2 is a large language model released by OpenAI based on Transformer Decoder architecture, which has the following characteristics [17].

First of all, GPT-2 is autoregressive, that is, it can focus on the already generated information of the previous text, making the generation of lyrics more coherent. Second, GPT-2 uses Transformer's powerful architecture, with each Decoder layer having a self-attention layer and a feed-forward neural network, as shown in Figure 3. At the self-attention level, the model learns the relationship between words and other words; In the feedforward neural network, the model processes each word independently. Among them, the self-attention layer uses multiple self-attention mechanisms that allow the model to focus on multiple locations at the same time, thus capturing different levels of information. This excellent structure allows our model to identify underlying stylistic and rhyming features in hip-hop lyrics.

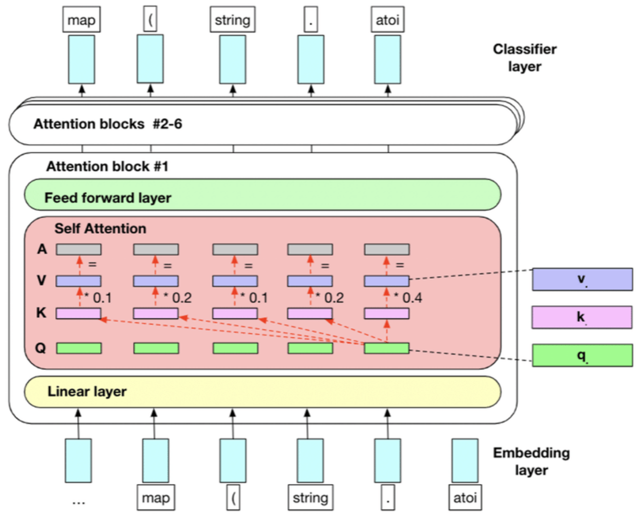


Figure 3. The overall structure of GPT-2

Using the structure of GPT-2 from Huggingface [18], the overall process of this model is as follows and shown in Figure 4:

1. Input a rapper's lyrics data (processed)
2. The tokenizer we selected processed the data and encoded it into a sequence form that the model could learn.
3. The model is trained on the input data, and the GPT-2's multi-headed attention mechanism and loss function carry out feature learning on the training data.
4. The lyrics generation function input prompt to the trained model and output the lyrics.
5. Three models were trained with the lyrics of three rappers (Drake, ASAP Rocky and Eminem), and model groups with three different rappers' characteristics were generated.
6. Lyrics will be generated via generation function and be passed to post processing if needed.

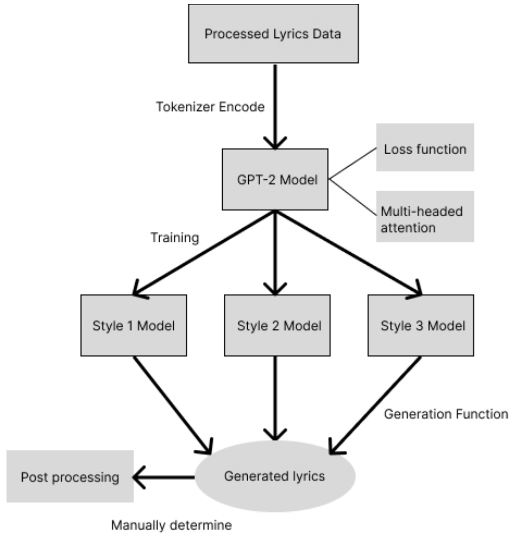


Figure 4. The workflow of the model

In order to better let the model, learn the rhyme features and the rationality of the lyrics content, we design the following characteristic modification.

First, we define the new loss function ourselves. We created a new class named RhymeModel that inherits the other methods of the GPT-2 class. Under the new class, we define the forward method ourselves. The overall loss is divided into two parts. The first part is cross entropy used by GPT-2, as shown in Formula 1, which is responsible for capturing the meaning of sentences. The second part is our own definition of rhyme-loss, calculating the rhyming rate of the predicted lyrics against the target lyrics, as shown in Formula 2. The final loss function is the sum of the two, and the two parts have their own weights, as shown in Formula 3. For the loss function we designed, we have the following thoughts. First, the loss function works by comparing the predicted sequence with the ground truth label sequence to calculate whether it rhymes. Due to the input characteristics of the loss function, the model calculates a single word every time. Therefore, such a loss function will take into account the features of global rhyme, including inter-sentence rhyme and end-of-sentence rhyme. In addition, the addition of weights is critical, and giving rhyme loss too much weight causes the model to go to the extreme of constantly generating the same words as prompt endings. Therefore, giving a high weighting to cross entropy gives the model a priority to generate complete and meaningful lyrics. We adopt the method of manual weight adjustment. We manually changed the weights through the lyric generation quality after each training.

$$L_{ce} = -\sum Q_i \ * \log(P_i)$$

Formula 1. Cross entropy loss (Q is the one-hot encoding of the real label, and P is the probability distribution predicted by the model)

$$L_{rhyme} = 1 - \frac{M}{N}$$

Formula 2. Rhyme loss (The total number of words is N, and the number of words that produce rhyme is M)

$$L = a * L_{ce} + (1 - a) * L_{rhyme}$$

Formula 3. Total loss (a is the weight)

Second, we call the tokenizer corresponding to GPT-2 provided by Huggingface [18]. The tokenizer helps us convert text data directly to the array understood by the model, namely encoding and decoding. At the same time, the tokenizer has the function of automatically trimming and padding the input data to an appropriate length and add a mask to the 0 padded to the data sequence so that the model can ignore their negative effects and avoid affecting the learning of the model.

Finally, we adjusted some hyper-parameters and trained the model to generate lyrics. We define a prompt, enter it into the pipeline function provided by Huggingface, and get the generated lyrics [18]. The Pipeline function does the following in this process:

1. Encode prompt is entered into the model as input
2. Let the model generate lyrics of a given length and output them through sequence data
3. Decode sequence output and convert it into text as output lyrics.

In this way, we successfully built the model, trained the model, and finally generated the text.

3.3 Post-train Filter

After directly generating the lyrics from our model, we also designed a post-train filter to improve the rhyme performance of our output. If the rhyme density of the lyrics does not reach our expectation, we can manually put the lyrics into our post-train functions to edit the ending of each line. The specific function of this filter is to replace the endings of the generated lyrics with new words which are rhymed and have similar meanings to the original ones. To achieve this goal, we used the pronouncing library to search for a list of words that rhyme with the given one [19]. Also, we imported the "gensim" library with pre-trained Glove Vectors, which utilizes the Word2Vec format for word embeddings and helps to calculate the cosine similarity between any two given words [20]. We obtained the pre-trained GloVe vectors from the Stanford NLP Group website [21], specifically employing the glove.6B.zip file in our code. This file includes vectors with dimensions of 50, 100, 200, and 300, trained on 6 billion tokens. Given that the gensim library employs the Word2Vec format for word embeddings, we're able to compute the cosine similarity between any two selected words after they've been converted to the Word2Vec format. A high similarity score suggests a substantial semantic correlation between the two words.

Utilizing this method, we can identify words from the list of rhymed words that have the highest similarity with the original final word of each line. Consequently, this allows us to maintain the original meaning while ensuring the lyrics rhyme to the highest possible degree. Finally, we divided the generated lyrics into two segments, each following a different rhyme word. Given that hip-hop lyrics may not consistently adhere to the same rhyme, we make the first half of the lyrics follow the rhyme of the final word in the first line, while the second half adopts the new rhyme of the last word in the middle line. This approach adds more diversity into our generated lyrics. The workflow of this post-train filter is shown in Figure 5. Some examples of our generated lyrics are shown latter in the appendix.

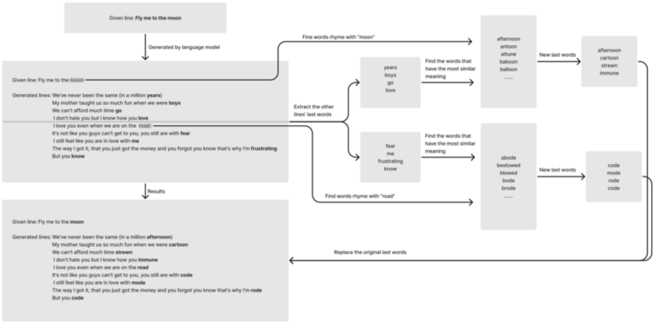

Given line: **Fly me to the moon**
Generated by language model
Given line: Fly me to the moon
Generated lines: We've never been the same (in a million **years**)
My mother taught us so much fun when we were **boys**
We can't afford much time **go**
I don't hate you but I know how you **love**
I love you even when we are on the road
It's not like you guys can't get to you, you still are with **fear**
I still feel like you are in love with **me**
The way I got it, that you just got the money and you forgot you know that's why I'm **frustrating**
But you **know**
Fine words rhyme with "moon"
Extract the other lines' last words
years
boys
go
love
fear
me
frustrating
know
Find the words that have the most similar meaning
Find the words that have the most similar meaning
afternoon
antoon
attune
baboon
balloon
......
abode
bestowed
blowed
bode
brode
......
Find words rhyme with "road"
New last words
New last words
afternoon
cartoon
strewn
immune
code
mode
rode
code
Results
Replace the original last words
Given line: Fly me to the **moon**
Generated lines: We've never been the same (in a million **afternoon**)
My mother taught us so much fun when we were **cartoon**
We can't afford much time **strewn**
I don't hate you but I know how you **immune**
I love you even when we are on the **road**
It's not like you guys can't get to you, you still are with **code**
I still feel like you are in love with **mode**
The way I got it, that you just got the money and you forgot you know that's why I'm **rode**
But you **code**

Figure 5. The workflow of post-train filter

4. Performance evaluation

4.1 Objective Evaluation

For the objective evaluation, we used functions to compute the rhyme density (RD) and perplexity (PPL) of the lyrics generated and the language model. Rhyme density, a measure used by Malmi et al., can validate the fluidity and proportion of rhymed lines [8]. Given that rhymed words in hip-hop lyrics can be both within the line or at its end, we determined the density by calculating the ratio of all the rhymed words in the lyrics. Perplexity serves as a measure of how well a probabilistic model can predict a sample and assess the language model. Higher value of perplexity implies that the language model has less certainty when predicting new words following the previous lines, so we expect a lower level of perplexity of our trained models. Since the GPT-2 model does not have a perplexity attribute, we opted to use a function to calculate our model's perplexity based on the loss. A model with lower perplexity is better to predict the test set and possesses greater certainty. Our results of the objective evaluation are in Table 1. We calculated the rhyme density of our generated lyrics for each style and compared them to the value of the original rappers' lyrics. We chose ten songs for each style and derived the average values. The results show that the rhyme densities of our generated lyrics are similar or higher than the original lyrics', this can imply that the rhyme performance of our lyrics is good. However, there are problems with such an outcome. From the generated lyrics, as shown in appendix, we can see that some parts of the verse are strictly rhymed and lose certain meaning, as well as multiple repeated sentences. Both of these situations increase the RD score, which may result in the model even having higher RD than rapper lyrics. At the same time, it also indicates that RD may be an overly general evaluation index, which will ignore some local problems.For perplexity, the values of our trained models are all higher than the original GPT-2 pre-trained model. This may be caused by the complexity we added to the loss function and the model, which can become the aspect we need to improve in the future. Besides, it is also likely that the post processing modifies the last word of lines such that the lines satisfies the RD but loses some meaning that increases PPL.

	Average RD	PPL
Trained from Drake	0.3933	326.86
Original Drake Lyrics	0.2614	(None)
Trained from ASAP Rocky	0.236	401.45
Original ASAP Rocky Lyrics	0.2499	(None)
Trained from Eminem	0.2808	815.46
Original Eminem Lyrics	0.2674	(None)
GPT-2 Pre-trained Model	(None)	162.47

Table 1. The results of objective evaluation

4.2 Subjective Evaluation

For the subjective evaluation, we designed a survey to invite participants to give feedback of our generated lyrics. We invited thirty-three participants who have the basic knowledge of hip-hop music to fill out our survey. The survey contained three questions which covered the evaluation of reasonability, rhyme performance, and style. For each question, the participants can rate on a scale of zero to ten, with a lower score indicating poorer performance. The results of the survey are generally positive. The average scores for the three questions are: 8.13, 7.56, and 6.78, which shows that the performance of reasonability and rhyme density are positive, while the style similarity should still be improved. Part of the results of the survey are also displayed in Figure 6, which shows the distribution of the attitudes of the participants for each question. We figured out that the attitudes are mainly distributed in the middle of the whole range, but the proportion of detractor is comparatively large for the style evaluation.

The results indicate that our raters are satisfied with the model in terms of rhyme and rationality, among which rhyme is more satisfied than rationality, but not so satisfied with style capture. We believe that this once again confirms the success of the loss function and post processing method under the muti-headed attention mechanism for GPT-2. However, post processing, while making the lyrics rhyme, undermines a degree of content consistency, which explains why reasonability score is lower than rhyme. For style capture, we only use the data of singers with different styles to train our model to achieve it, which shows that our method in this aspect lacks sufficient effectiveness.

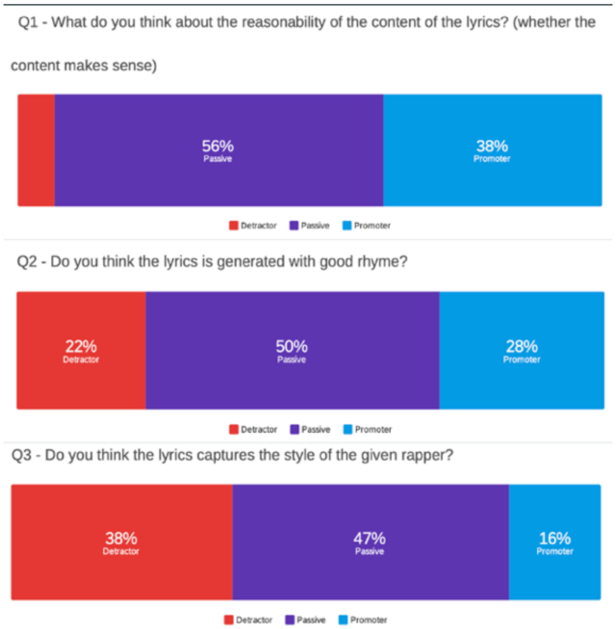


Figure 6. Part of the survey results

4.3 Evaluation Conclusion

To sum up, both objective and subjective perspectives reflect some common problems and satisfactory results. We had good success with rhyme and content balance, as demonstrated by RD and reasonability scores. However, the scoring dimension and the application of various methods of the model, such as the application of post processing, are insufficient. This means that we need to improve these two perspectives in the future.

5. Conclusion and future work

In conclusion, the model successfully fulfilled the project's goal of exploring a hip hop style lyric generator that combines both rationality and rhyming.

Firstly, we adopted GPT-2 model with multi-headed attention characteristics as our model framework. On this basis, we modify the loss function of the model and define a new compound loss function by adding a new rhyme loss and weight on the basis of the general cross entropy of the model. For the generated lyrics, we also have the method of post processing. By searching the words at the end of each lyric with the same meaning but rhyming with the previous lyrics, we select an optimal word to modify the word at the end of the lyric sentence. In this way, we guarantee a certain degree of unchanging meaning and guaranteed rhyming lyric generation in the event that we find the generated lyrics unsatisfactory. In terms of data, we made sure that the data presented strong features by cleaning repeated lines, extraneous content and retaining rhyming lines. At the same time, we extracted data sets of three rappers with different styles and trained three models separately to ensure different styles of learning.

We believe that GPT-2's multi-headed attention, data preprocessing and our loss function jointly promote GPT-2 to capture the content of lyrics, rapper style and prorhythmic features. A balanced generation of hip-hop lyrics that we wanted. We believe that this is an attempt to apply deep learning to the field of music, and such an attempt will benefit the creative efficiency and creative possibilities of hip-hop music.

In the future, we hope to make improvements in the following areas:

1. We hope that loss function is more complex and has more parts to capture more features, such as style capture, so that loss function can clearly and strictly regulate the learning of features. At present, loss function is only limited to the reflection of two features, and the definition of rhyme loss is simply a density calculation.
2. Data sets can be expanded. This helps the model to further learn different features, because the lyrics we have generated so far are not perfect, and we think the number of lyrics will help improve the quality of the model.
3. Parameters can be optimized in a better way, such as finding a more appropriate weight definition of loss function. At present, the definition of some parameters, especially the weight part, adopts manual adjustment, by trying different combinations and objective evaluation results to reflect the pros and cons.
4. Find a more intelligent way to conduct post processing, instead of deciding whether to conduct it manually, and a better methodology to conduct post processing. At present, post processing is a method to complete the insufficient lyrics generated by the model. We only need to manually judge whether it is necessary or not. Meanwhile, such modification will destroy the meaning of the sentence to a certain extent.

References

[1] C. L. Keyes, “UI Press | Cheryl L. Keyes | Rap Music and Street Consciousness.” https://www.press.uillinois.edu/books/?id=p072017 (accessed May 18, 2023).

[2] F. B. Krohn and F. L. Suazo, “CONTEMPORARY URRAN MUSIC: Controversial Messages in Hip-Hop and Rap Lyrics,” *ETC Rev. Gen. Semant.*, vol. 52, no. 2, pp. 139–154, 1995.

[3] Qiang Yang, Yang Liu, Tianjian Chen, and Yongxin Tong, “Federated Machine Learning: Concept and Applications: ACM Transactions on Intelligent Systems and Technology: Vol 10, No 2.” https://dl.acm.org/doi/10.1145/3298981 (accessed May 18, 2023).

[4] Z. Obermeyer and E. J. Emanuel, “Predicting the Future — Big Data, Machine Learning, and Clinical Medicine,” *N. Engl. J. Med.*, vol. 375, no. 13, pp. 1216–1219, Sep. 2016, doi: 10.1056/NEJMp1606181.

[5] M. Hernandez, S. Kahane, S. Fleury, and K. Gerdes, “Automatic Generation of Hip-Hop and Rap Lyrics”.

[6] A. Vaswani *et al.*, “Attention is All you Need,” in *Advances in Neural Information Processing Systems*, Curran Associates, Inc., 2017. Accessed: May 18, 2023. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html

[7] L. Xue *et al.*, “DeepRapper: Neural Rap Generation with Rhyme and Rhythm Modeling.” arXiv, Jul. 05, 2021. doi: 10.48550/arXiv.2107.01875.

[8] Eric Malmi, Pyry Takala, Hannu Toivonen, Tapani Raiko, and Aristides Gionis, “DopeLearning | Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining.” https://dl.acm.org/doi/10.1145/2939672.2939679 (accessed May 18, 2023).

[9] Y. Bai, J. Mei, A. Yuille, and C. Xie, “Are Transformers More Robust Than CNNs?” arXiv, Nov. 09, 2021. doi: 10.48550/arXiv.2111.05464.

[10] J. Maurício, I. Domingues, and J. Bernardino, “Comparing Vision Transformers and Convolutional Neural Networks for Image Classification: A Literature Review,” *Appl. Sci.*, vol. 13, no. 9, Art. no. 9, Jan. 2023, doi: 10.3390/app13095521.

[11] Peter Potash, Alexey Romanov, and Anna Rumshisky, “GhostWriter: Using an LSTM for Automatic Rap Lyric Generation - ACL Anthology.” https://aclanthology.org/D15-1221/ (accessed May 18, 2023).

[12] Nikola I. Nikolov, Eric Malmi, Curtis Northcutt, and Loreto Parisi, “Rapformer: Conditional Rap Lyrics Generation with Denoising Autoencoders - ACL Anthology.” https://aclanthology.org/2020.inlg-1.42/ (accessed May 18, 2023).

[13] “Genius | Song Lyrics & Knowledge.” https://genius.com/ (accessed Apr. 07, 2023).

[14] “Beautiful Soup Documentation — Beautiful Soup 4.12.0 documentation.” https://www.crummy.com/software/BeautifulSoup/bs4/doc/ (accessed Apr. 21, 2023).

[15] emakpati, “How to Collect Song Lyrics with Python,” *Medium*, Dec. 09, 2020. https://towardsdatascience.com/song-lyrics-genius-api-dcc2819c29 (accessed May 18, 2023).

[16] S. M. Lakew, M. Cettolo, and M. Federico, “A Comparison of Transformer and Recurrent Neural Networks on Multilingual Neural Machine Translation,” in *Proceedings of the 27th International Conference on Computational Linguistics*, Santa Fe, New Mexico, USA: Association for Computational Linguistics, Aug. 2018, pp. 641–652. Accessed: May 18, 2023. [Online]. Available: https://aclanthology.org/C18-1054

[17] “OpenAI GPT2.” https://huggingface.co/docs/transformers/model_doc/gpt2 (accessed May 11, 2023).

[18] “Models - Hugging Face.” https://huggingface.co/models (accessed Apr. 07, 2023).

[19] A. Parrish, “pronouncing: A simple interface for the CMU pronouncing dictionary.” Accessed: Apr. 21, 2023. [Online]. Available: https://github.com/aparrish/pronouncingpy

[20] R. Rehurek, “gensim: Python framework for fast Vector Space Modelling.” Accessed: Apr. 21, 2023. [OS Independent]. Available: http://radimrehurek.com/gensim

[21] “GloVe: Global Vectors for Word Representation.” https://nlp.stanford.edu/projects/glove/ (accessed Apr. 21, 2023).

Appendix

A. Sample Verses

(1) Trained from Drake’s lyrics:
She likes to ride in the AMs in my cool whip
I was really into the niggas that was in the ship
The Niggas were always grip
They don't tell me, the kids, 'Hey, I said so, you know ship
You're the star, don't forget tip
It's why I'm always the star
My name's Amar
I love superstar
I told them my name was Amar
She told me I was a rapper, I was ribald
I just told myself I should've stayed away from AR

(2) Trained from A$AP Rocky’s lyrics:
I'm just laying on the floor again, trying to find my partner
'Cause you couldn't care less about the same shit the same time
It's the old days, where the young black male just stepped up
He can't get ahold of a name, let alone a piece of shit to show for himself

And don't put any hustle on me, I'm just showing off my skills
I'mma get you some mills
You bitches bitches bitching that these little bitches don't ills
Bitch, I'm the new mom, you little stills

(3) Trained from Eminem's lyrics:
I woke up like this, and I'm not even looking
Momma, my purse's cold, my purse's cold, my purse's cold, my purse is scared
Go figure, I don't really make a living
I work too hard, I take more shit from you
Why don't you just let me? You can't let the dog go to sleep without asking
I'm from the East, I'll take what I can, but I won't hold back