PIC2RAP:IMAGE-INSPIRED STYLISTIC HIP-HOP LYRICS GENERATOR

by

Boqing Zheng

Signature Work Product, in partial fulfillment of the
Duke Kunshan University Undergraduate Degree Program

March 7, 2024

Signature Work Program
Duke Kunshan University

APPROVALS

Mentor: Dongmian Zou, Division of Natural and Applied Sciences

---

Marcia B. France, Dean of Undergraduate Studies

CONTENTS

Abstract ii

Acknowledgements iii

List of Figures iv

List of Tables v

1 Introduction 1

2 Material and Methods 5

3 Results 14

4 Discussion 17

5 Conclusions 25

References 27

A Appendix A 29

B Appendix B 32

ABSTRACT

Since its inception, hip-hop has gained global attention for its unique cultural symbolism and insightful social commentary. As an art form, hip-hop expresses emotions and opinions through beats, rhymes and wordplay. With the development of artificial intelligence and deep learning technologies, it has become possible to explore the use of machine learning to create hip-hop lyrics with the goal of capturing its unique style. However, automatically generated lyrics often lack the flow, coherence, and style of hip-hop. Considering that hip-hop compositions are inspired not only by words but also by images, existing models fail to fulfill this need. To this end, we develop a novel automatic hip-hop lyrics generation model that combines image recognition techniques and linguistic generation models with the aim of creating lyrics that reflect the spirit of hip-hop. The model is based on VisionEncoderDecoderModel and GPT-2, and realizes the conversion from vision to text by recognizing the image content and then creating hip-hop lyrics in the style of the featured artist. The image recognition model is pre-trained, while the language model is fine-tuned by training an artist-specific lyrics dataset to learn language patterns and rhyme structures. Rhyme density and coherence are optimized using a specific processing function and an improved loss function to ensure the quality of the lyrics. The model performs well in terms of rhyme density, coherence, and stylistic consistency, and its performance is validated by both objective and subjective evaluations, advancing the technology of automated hip-hop lyrics generation and exploring new ways to combine music and visual arts.

ACKNOWLEDGEMENTS

I would like to give special thanks to our mentors and partners, without whose support and help this project would not have been possible.

First of all, I would like to express my sincere gratitude to my supervisor, Prof. Dongmian Zou, for his invaluable guidance and relentless support at every step of the project, from its inception to its completion. Prof. Dongmian Zou not only guided us in the inspiration of the project, but also provided in-depth expertise and advice in the implementation of the model's multimodal task and the model optimization process.

At the same time, I would like to thank Prof. Peng Sun for his careful guidance and extremely critical technical support in the development of the lyric generation model and the optimization methodology. Prof. Peng Sun's rigorous research attitude and superb professional skills were decisive for the smooth progress of our research.

In addition, I would like to thank my colleague Shuhe Wang for her great help in the data processing and optimization of the lyric generation model, whose diligence and professionalism greatly advanced the research progress and positively impacted the teamwork.

Meanwhile, I need to thank the questionnaire participants in the evaluation phase of this project, whose participation and feedback were crucial to our research.

LIST OF FIGURES

1.1 The sample lyrics from Hernandez et al.[9] . . . 3

2.1 Part of the cleaned dataset . . . 6
2.2 Overall structure of the model . . . 6
2.3 Structure of VisionEncoderDecoderModel . . . 7
2.4 Structure of GPT-2 . . . 8
2.5 The workflow of the lyrics generation model . . . 9
2.6 GPT-2 Tokenizer example . . . 11
2.7 Word2Vec Example (arrows represent the similarity) . . . 12
2.8 Workflow of the post-training filter . . . 13

3.1 Image and caption generated by VisionEncoderDecoder model . . . 15
3.2 Example of generated lyrics based on a given prompt. . . . 16

4.1 Survey results Q1 . . . 22
4.2 Survey results Q2 . . . 22
4.3 Survey results Q3 . . . 23

A.1 Image and caption generated by VisionEncoderDecoder model example 1 . . . 29
A.2 Image and caption generated by VisionEncoderDecoder model example 2 . . . 30
A.3 Image and caption generated by VisionEncoderDecoder model example 3 . . . 30
A.4 Image and caption generated by VisionEncoderDecoder model example 4 . . . 31
A.5 Image and caption generated by VisionEncoderDecoder model example 5 . . . 31

LIST OF TABLES

4.1 Comparison of CIDEr Scores across different models on the COCO dataset. . . . 19
4.2 Rhyme Density (RD) and Perplexity (PPL) for generated and original lyrics . . . 20

Chapter 1

INTRODUCTION

Since the 1970s, hip-hop music has played an increasingly prominent role in youth culture around the world, attracting an enthusiastic following of young people. Performers of this form of music, hip-hop artists, often use the medium of song to convey their insights and personal experiences, which move listeners and inspire strong emotional resonance. In this genre, the content and message of the lyrics are far more important than the construction of complex melodies, and have become a central element in the creation of rap music[11]. Most prominent hip-hop performers have their own unique style of songwriting, because hip-hop songwriting has to strike a balance [11] between the dimensions of rhyme matching, wordplay, fluency, logical coherence, emotional expression, and deeper meaning. At the same time, a large number of interviews and studies have revealed how hip-hop artists use visual images to express their inspiration and vision for their creations [8], and many cases of research have demonstrated that musical compositions are often rooted in the interpretation of visual images [4]. This phenomenon suggests that there is a strong connection between visual elements and hip-hop music, and that it is particularly important to utilize specific visual images to assist hip-hop artists in their songwriting. Therefore, in this context of artistic creation, our project focuses on inspiring hip-hop artists with the help of images in order to create high-quality lyrics with profound connotations.

In recent years, with the rise and advancement of machine learning and deep learning technologies, these advanced methods have enabled computational models to learn and predict algorithms from existing data, which in turn independently generate new lyrics[18]. Given that hip-hop lyrics have their own inherent structural and stylistic norms, the use of machine learning techniques to automatically generate hip-hop lyrics has become a popular topic among researchers. In addition, these techniques have been widely applied to challenges in the field of image description; these image description models are capable of automatically generating natural language descriptions based on the content of an image, and they seamlessly combine knowledge from computer vision and natural language processing. All these advances have proved that utilizing images to create hip-hop lyrics is completely feasible on a technical level.

Previous studies have adeptly harnessed the capabilities of machine learning to deconstruct

and comprehend the intricate structures characteristic of rap lyrics, subsequently utilizing these insights to generate novel lyrical content. Malmi et al. pioneered this domain with the development of a predictive algorithm known as DeepBeat, which employed the RankSVM algorithm alongside sophisticated deep neural network models to craft subsequent lines of lyrics to complement pre-existing ones [29]. Building upon this foundation, Hernandez et al. refined this approach through the application of Word2Vec to "ascertain the optimal window size, word embeddings, and to generate versified content," thereby enhancing the syntactic precision and fluency of the generated lyrical output [9].

More recently, Xue and colleagues have innovated further by designing a Transformer-based system to produce rap lyrics, culminating in the creation of a model dubbed DeepRapper. This model boasts the advanced capability to simultaneously generate lyrics infused with rhythm and rhyme [29]. In our endeavor, we are electing to forge a Transformer-based model specifically tailored for Natural Language Processing (NLP) and intricate lexical associations. The Transformer model's architecture is notably proficient at assigning weights to each word in a sequence based on its relevance relative to other words, capturing long-range dependencies and contextual nuances more effectively than its technological predecessors [25]. In comparison to Convolutional Neural Networks (CNNs), Transformers possess a superior capacity to contemplate global contextual information [2]. Maurício et al. have also acknowledged that Transformer-based architectures can achieve heightened levels of accuracy and efficiency in NLP tasks [14].

The preeminent challenge within the current methodologies lies in balancing the precision of rhyme schemes with the logical coherence and intrinsic quality of the lyrics. Former systems devised for rap lyric generation either incorporated a unique tag at the end of rap verses to facilitate the learning process [15] or employed a "two-phase strategy" for rhyme modeling, which entailed generating the rap lyrics first and subsequently appending rhyme schemes [22]. However, outcomes from prior research indicate that lyrics generated via these methods often lack fluidity and effectiveness. Moreover, the models struggle to fully assimilate rhyme schemes, resulting in a paucity of diversity in rhyming terminations. For instance, as depicted in Figure 1.1, the Word2Vec methodology appeared proficient at producing rhyming lyrics yet faltered in maintaining the lyrical cohesiveness [9]. Attempts by Xue et al. and Malmi et al. to design novel training procedures aimed at teaching models to learn and produce rhyming verses resulted in outputs that were still perceived as monotonous and simplistic in style and structure [29][13].

Thus, our motivation is to seek methodologies that refine and enhance the existing models, with the objective to augment the creativity, rhythmic diversity, and cogency of the generated rap lyrics. An endeavor scarcely explored is the stylized generation of rap lyrics emulating the distinct styles of renowned hip-hop artists. Hence, our ambition is to architect a language model capable of engendering high-caliber hip-hop lyrics that embody specific stylistic attributes.

In the model design phase of this study, we used a fusion approach that combines a pre-trained visual encoder-decoder model for image description with a fine-tuned language model based

INPUT	*To the left*	
SYLLABLES	Thirteen syllables per bar	Seven syllables per bar
RHYME SCHEME	AAAA BBBB C DD	AAA BBB CCC DD
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		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

Figure 1.1: The sample lyrics from Hernandez et al.[9]

on the Transformer architecture to generate creative lyrics. Specifically, for the task of image description, we chose the Visual Transformer (ViT) as the encoder, an encoder capable of capturing the complex features of an image and transforming them into a high-level representational form that can be processed. Thereafter, GPT-2, a powerful language model, is used as a decoder to generate relevant descriptive text based on the image content. This codec structure has proven its effectiveness in the image caption creation task, not only in understanding the visual information of the image, but also in expressing this information precisely in natural language.

In a further step of generating lyrics, we pre-trained the GPT-2 model [19] and fine-tuned it so that it not only generates coherent text, but also creates rhyming lyrics. We experimented with two rhyming techniques that are common in previous studies: one is to ensure end-of-line rhymes through post-processing filters after the lyrics have been generated, and the other is to train the model to directly learn and produce rhyming lyric structures[29]. Also, we innovatively adjusted the loss function of the model so that it can better capture and learn rhyming features.

In order to make the generated lyrics style-specific, we chose lyrics from specific rap artists as training data. The goal of this approach is to generate lyrics that are not only technically rhyme- and rhythm-compliant, but also stylistically similar to the original artist's work.

For our VisionEncoderDecoderModel, we took an objective evaluation approach to measure its performance. We used the CIDEr (Consensus-based Image Description Evaluation) metric and compared VisionEncoderDecoderModel with other peer models. However, given that the assessment of lyrics quality usually involves subjective taste and emotional resonance, we thoroughly evaluated the lyrics generation model on both objective and subjective dimensions. For the objective assessment, we measured the quality of the generated lyrics based on metrics such as rhyme density and perplexity. For subjective assessment, we conducted a human participatory survey in which participants evaluated the generated lyrics on multiple dimensions such as fluency, creativity, emotional expression, and stylistic mimicry.

In the subsequent sections, we will elaborate our research methodology, technical path and evaluation system based on existing studies. Ultimately, we will analyze the model performance in depth, summarize the key findings, and present useful suggestions for future work and potential improvements.

Chapter 2

MATERIAL AND METHODS

2.1 Dataset Preparation

We targeted specific rap artists for lyrics collection and chose Genius.com as the primary data source, which stands out for its large and diverse lyrics database [7]. To extract the lyrics, we utilized the BeautifulSoup tool, and with the client access token provided by Genius.com, we obtained the artist-related lyrics information through an API interface and returned it in json format [3] [5]. The code we developed is able to recognize the songs of a given rap artist on Genius.com and crawl the corresponding lyrics. We can also preset the number of songs extracted for each artist and save them as a text file sorted by popularity. So far, we have successfully collected a large number of lyrics totaling thousands of lines from Drake, ASAP Rocky and Eminem, three famous American rap artists.

During the data cleaning process, we remove all tags such as ”[Verse]” and ”[Bridge]”, as well as extra blank lines between verses, and also remove textual impurities such as punctuation marks. Such treatment makes it easier for the model to capture the semantic connections between words. However, in our experiments, we found that the model had a tendency to generate repetitive passages of lyrics, as these passages performed better in terms of rhyme and coherence. For this reason, we again eliminated repeated sentences from the dataset. In order to enhance the model's learning of rhyming techniques, we attempted to train with only rhyming sentences. The cleaned data samples are displayed in Figure 2.1.

2.2 Model Design

In our project, we adopt an innovative approach to generate lyrics based on image content. Our framework consists of two parts: firstly, the VisionEncoderDecoderModel, which is used to parse the images and provide descriptive text, and secondly, the GPT-2 based lyrics creation model, which utilizes the output of the former as a source of cue-word inputs for creating lyrics. The benefit of this splicing model is that it is able to directly translate visual information into textual descriptions, which are then used to guide the creation of lyrics, resulting in lyrics that are closely related to the content of the image. Such a process mimics the natural process of

```
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 2.1: Part of the cleaned dataset

observation, interpretation and creation by human artists, making the connection between lyrics and images more vivid and meaningful. The overall structure of our model is presented in Figure 2.2

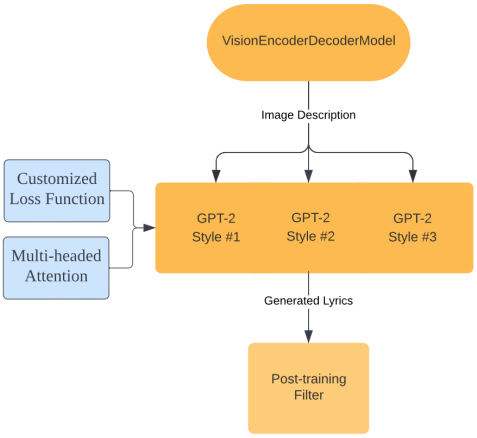


Figure 2.2: Overall structure of the model

2.2.1 Image Description Model

In this project, we use a state-of-the-art image captioning generation model that incorporates computer vision and natural language processing techniques, specifically, it employs the VisionEncoderDecoderModel structure, implemented by the pre-trained nlpconnect/vit-gpt2-image-captioning model. This model combines the Vision Transformer (ViT) as an encoder, and the GPT-2 as a decoder to encode image content into visual features and decode these features into natural language descriptions [10].

We chose to use a pre-trained VisionEncoderDecoderModel for our project mainly because it saves time and computational resources while ensuring model performance. This pre-trained model has been trained on a large and diverse dataset, so it is able to extract key features from images quickly and efficiently. If we had chosen to train the model from scratch on our own, we would have faced a huge investment in time and resources, and would have needed a large

amount of high-quality training data. In addition, the pre-trained model has been validated in multiple tasks and environments, which ensures high performance and stability [10]. While self-training allows for greater customization, the efficiency and utility of pre-trained models often make them the superior choice.

In terms of technical details, the ViTFeatureExtractor is used to process the input image and extract a feature representation suitable for processing by the model. This step includes a series of preprocessing operations such as image resizing, normalization, etc. to fit the input requirements of the ViT model, which is a neural network based on a self-attention mechanism that divides an image into a series of small patches (patches), which are then linearized and processed through a series of self-attention layers in order to capture the relationships between the different blocks of images [6].

The processed image features are fed into the VisionEncoderDecoderModel, where the encoder part, i.e., ViT, converts these features into intermediate hidden states, which are then passed on to the decoder part, GPT-2. GPT-2 is an autoregressive model that is capable of generating sequential text sequences based on the context provided by the encoder [19]. In this scenario, GPT-2 progressively generates a description of the image, one word at a time, until it reaches the end of the sentence. The generated text sequence is then decoded into human-readable text by the GPT2Tokenizer. The tokenizer converts the digital sequences output by the model into corresponding words or tokens while removing any special tokens, such as end-of-sequence tokens, to produce clean captioned text. In this process, images are converted from pixel values to text describing their contents through a highly coordinated encoding-decoding process, which is a typical example of deep learning cross-modal transformation. The overall structure of our image description model is shown in Figure 2.3.

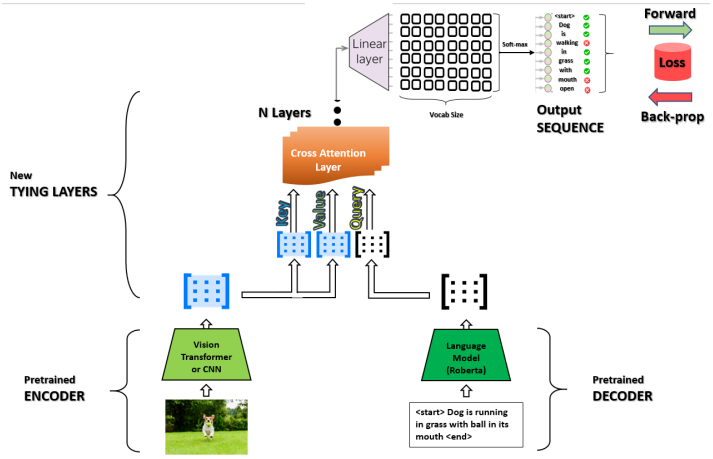


Figure 2.3: Structure of VisionEncoderDecoderModel

2.2.2 Lyrics Generation Model

In our project, the core model is trained using the state-of-the-art architecture of GPT-2, a generative pre-trained model that performs excellently in natural language processing tasks [12]. GPT-2 is based on the Transformer model, whose most notable feature is the utilization of the

self-attention mechanism, which allows it to excellently capture and utilize global contextual information when dealing with the task of generating lyrics [25]. This is essential to ensure that the generated lyrics are semantically continuous and intrinsically logically tight.

The GPT-2 model, published by OpenAI, is unique in building language models. It is autoregressive, meaning that the model is able to take into account what has already been generated in the previous text when generating new text segments, resulting in coherent and logically tight lyrics[19]. The internal structure of GPT-2 contains multiple decoder layers, each consisting of a self-attention layer and a feed-forward neural network, as shown in Figure 2.4. In the self-attention layer, the model understands the interactions and meanings of different words by learning the relationships between words, while in the feed-forward neural network, each word is processed independently, which ensures the model's accuracy and efficiency in processing each word.

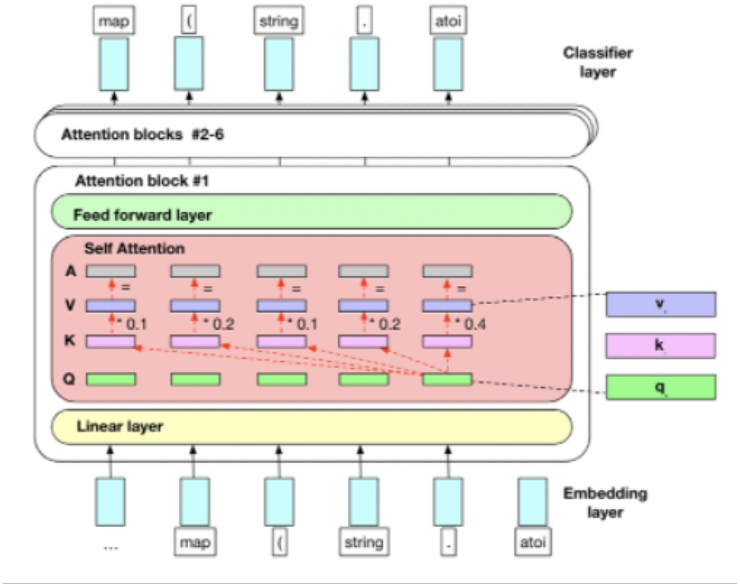


Figure 2.4: Structure of GPT-2

In addition, GPT-2's multi-head self-attention mechanism allows the model to focus on multiple locations in the text at the same time, which is particularly useful for capturing complex structures and rhyming patterns in hip-hop lyrics. This ability not only helps the model to recognize and learn the unique style and rhythm of hip-hop music, but also to better mimic the creative style of real artists when generating new lyrics, making the generated content both novel and artistic. The combination of these features made the GPT-2 an ideal choice for our project, and with its powerful functionality as a foundation, our model was able to create lyrics that were both deep and creative.

In this project, we adopt the GPT-2 architecture provided by the Huggingface community as the base model in order to build a deep learning system specialized for rap lyrics generation [19]. The entire workflow of model construction and training is shown in Figure 2.5, and the detailed steps include: First, the model receives preprocessed lyrics data from rappers. These data have been carefully screened and processed for subsequent training requirements. Second,

we employ a specially selected tokenization tool to process the raw text data. The tokenizer not only breaks sentences and separates words, but also transforms the text into a sequence-encoded form that the model can understand and learn. Next, we trained the model on the processed data. At this stage, the GPT-2 model effectively learned the deep features of the lyrics data through loss function optimization with the help of its built-in multi-head attention mechanism. Subsequently, by providing specific cues (e.g., the first few words of the lyrics) to the trained model, the model was able to automatically continue writing the complete lyrics. In addition, we trained three separate models for three different styles of rappers-Drake, ASAP Rocky, and Eminem. This is done to capture the unique creative style of each artist and reflect it in the lyrics generation. Finally, the generated lyrics are output through a set of customized generative functions and, if necessary, a post-processing step to optimize the quality of the generated results. Through this process, our model is able to create personalized lyrics with a specific artistic style.

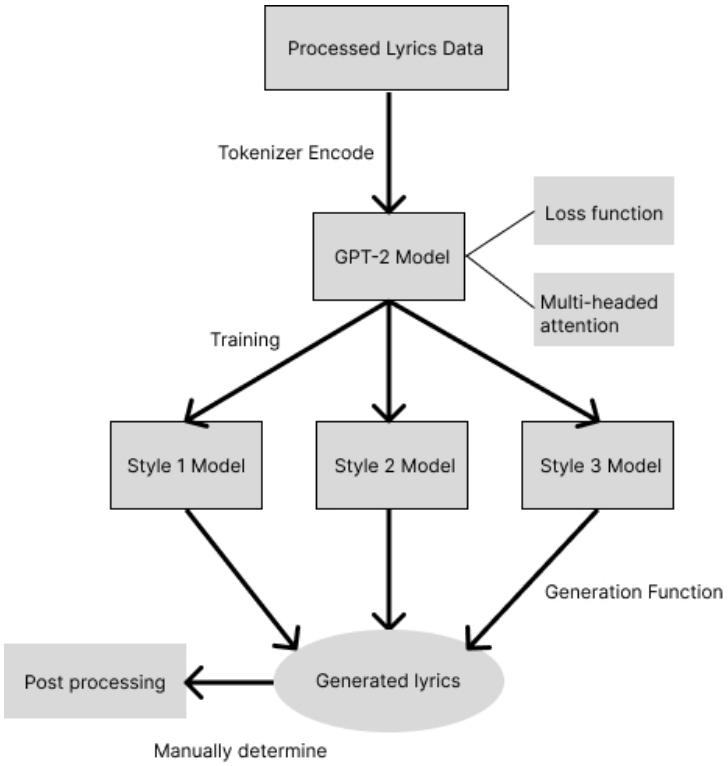


Figure 2.5: The workflow of the lyrics generation model

In order to optimize the model's learning effect on the rhyming properties of lyrics and their semantic coherence, we make innovative improvements to the loss function during model training. Specifically, we construct a new category called RhymeModel, which inherits the original method of GPT-2 and incorporates our customized forward propagation algorithm.

In this function, the total loss consists of two parts: the first part maintains the original cross-entropy loss of the GPT-2 model, which is responsible for capturing the basic semantic information of a sentence to ensure the coherence and meaning integrity of the content of the generated lyrics. The cross-entropy loss is expressed in Formula 2.1, where Q is the one-hot encoding of the real label, and P is the probability distribution predicted by the model.

$$L_{ce} = -\sum Q_i \cdot \log(P_i) \tag{2.1}$$

The second part is our ad-hoc rhyming loss, which is designed to quantify the rhyming similarity between the model's predicted outputs and the target lyrics, and thus to enhance the model's expressive power in terms of rhyming. The rhyme loss is expressed in Formula 2.2, where the total number of words is N, and the number of words that produce rhyme is M.

$$L_{rhyme} = 1 - \frac{M}{N} \tag{2.2}$$

The final loss function is then a weighted sum of these two components, where the weights of each component are carefully designed to ensure that the model is able to balance semantic understanding and rhyming skills during the learning process. The final loss is expressed in Formula 2.3, where a is the weight.

$$L = a \cdot L_{ce} + (1 - a) \cdot L_{rhyme} \tag{2.3}$$

In the design of the loss function, we focus on the following aspects: first, the loss function performs its function by comparing the degree of rhyme matching between the predicted sequence and the real labeled sequence, and due to the characteristics of the inputs, this loss function only processes the rhyme similarity between the next individual word it predicts and the target lyrics in each iteration, and this design allows the model to take into account the global rhyming patterns in the lyrics, including cross-stanza rhyming and end-of-stanza rhyming, among other scenarios. Second, the setting of the weights is crucial, as too high a weight on rhyme loss may cause the model to favor the generation of rhymes at the expense of semantics, and conversely, higher cross-entropy weights help to ensure the completeness and comprehensibility of the generated content. In practice, we adopt a strategy of manually adjusting the weights to fine-tune the weights according to the quality of the generated lyrics after each training session in order to achieve the best training results.

In our project, we utilize a GPT-2 architecture-matched tokenizer tool provided by the Huggingface community, whose core function is to convert raw textual data into numerical sequences that can be parsed by the model, i.e., to perform encoding and decoding operations [19], as shown in Figure 2.6. In addition, the tokenizer can automatically trim and fill the input data to ensure that all data sequences are of the same length, and for the filled parts of the sequences, the tokenizer will add the corresponding mask information so that the model can ignore these non-substantive information during the learning process, thus avoiding them from adversely affecting the learning effect.

In the training phase of the model, we carefully tune several hyperparameters to optimize the model's performance in lyric generation. We designed a specific cue message and fed it into the model via the model pipeline function provided by Huggingface [19]. The pipeline function undertakes the following key tasks in this sequence of operations. First, it feeds the encoded

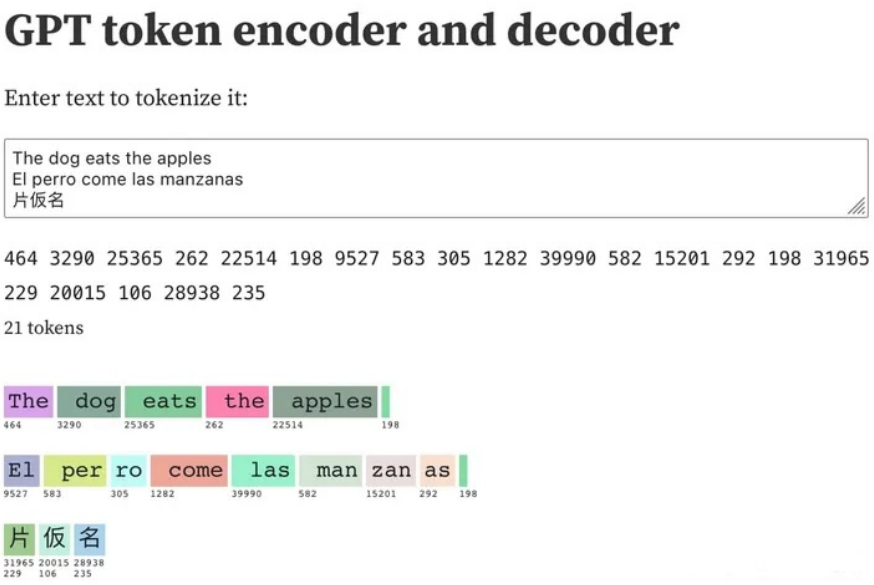


Figure 2.6: GPT-2 Tokenizer example

cue message we set as input to the pretrained model. Second, it directs the model to generate sequences of lyrics of a certain length based on the input cues, and uses these sequences as output. Finally, it decodes the output of the model's sequences into human-understandable text form, and finally obtains the content of the generated lyrics. Through the above process, we not only build and train an efficient model, but also realize the automated generation from text cues to lyrics text.

2.3 Post-training Filter

In order to further refine the lyrics generated by the model and to enhance their rhyming effect, this study introduces an innovative post-training filter at the model output stage. The idea behind the design of this filter is to manually intervene after generating lyrics that fail to satisfy the established criteria for the rhyming density of the lyrics by feeding the lyrics into a specialized processing function. In this function, we scrutinize and modify the endings of each line of the generated lyrics to match the rhyming structure and be semantically consistent. In addition, to ensure that the replaced new words both rhyme and are similar in meaning to the original text, we rely on the pronouncing library [20], which provides a rich library of rhyming words that can effectively match and replace rhyming words.

In addition, this project introduces a pre-trained GloVe word vector to enhance the judgment of semantic similarity. By integrating the "gensim" library, we are able to embed words in Word2Vec format and compute the cosine similarity between any two words [24], as exemplified in Figure 2.8. We obtained these pre-trained GloVe vectors from resources provided by the Natural Language Processing Group at Stanford University [21], and used the glove.6B.zip file, which contains multi-dimensional vectors, in our code implementation. These vectors have been trained on a large-scale corpus and thus are able to accurately capture the seman-

tic links between words. With the help of the gensim library to process word embeddings in Word2Vec format, we can quantitatively assess the semantic proximity between two words. Word pairs with high similarity scores indicate strong semantic relatedness, which is crucial to ensure that the lyrics will not only rhyme but also be coherent after modification. Through this series of well-designed post-processing steps, we have significantly improved the quality of the generated lyrics, making them more artistically appealing.

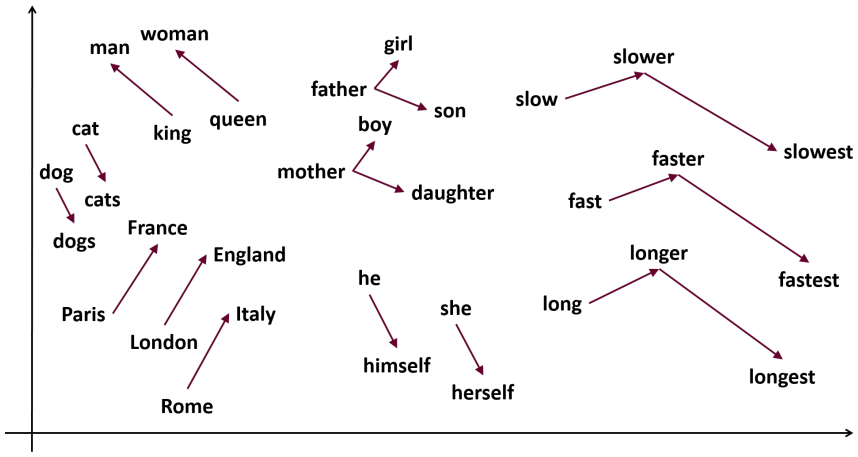


Figure 2.7: Word2Vec Example (arrows represent the similarity)

By adopting this unique strategy, we are able to accurately filter out the rhyming words from the pre-constructed rhyming lexicon that most closely match the semantic meaning of the words at the end of the original lyrics. This process not only carefully maintains the original intent and emotional color of the lyrics, but also maximizes the artistic appeal of the lyrics through accurate rhyme matching. In the final stage of the process, we divided the generated lyrics into two separate parts, each ending with a selected rhyming word. Considering the variability of rhymes in hip-hop songwriting, we cleverly designed the lyrics structure so that the first part of the lyrics continued the rhymes of the opening lines, while the second part shifted to the new rhymes presented by the end words of a line in the middle. This clever structuring not only gave our generated lyrics more variety, but also made the entire lyrics more colorful in terms of rhyme and rhythm. Figure 2.8 exhaustively depicts the flowchart of the operation of the post-training filter we developed. And in order to demonstrate the effectiveness of this approach more intuitively, we provide a series of samples of actual generated lyrics in the Appendix B.

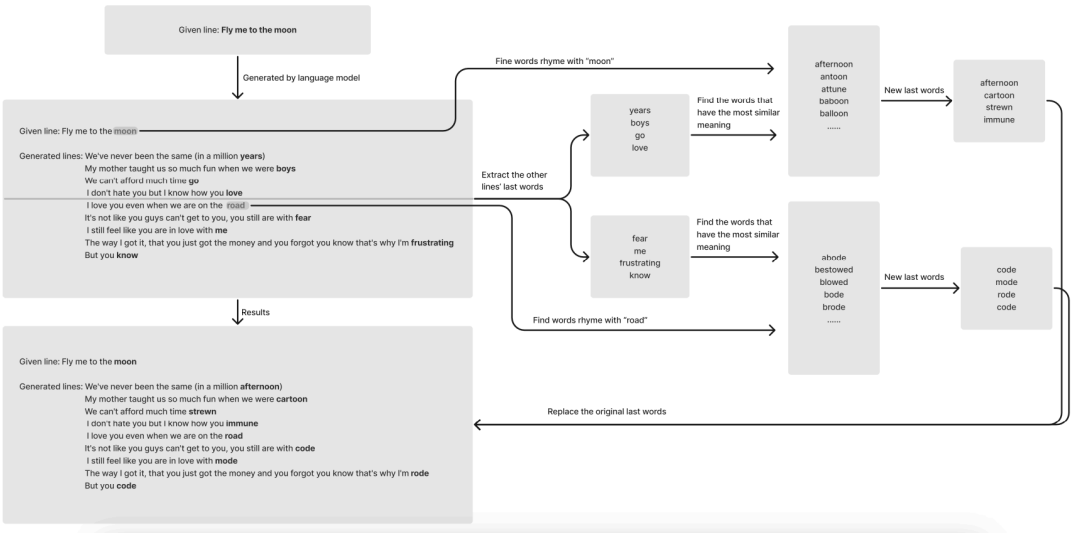


Figure 2.8: Workflow of the post-training filter

Chapter 3

RESULTS

In the results section of this study, we perform a comprehensive series of result-generating simulations of the developed vision-linguistic model and verify its ability to automatically generate image descriptions in a later section. First, we had the image description model generate a series of descriptions of images. These generations utilized the Python programming language and combined the widely used COCO 2017 validation dataset provided by TensorFlow Datasets (TFDS) with the VisionEncoderDecoderModel, ViTFeatureExtractor, and GPT2Tokenizer developed by Huggingface and GPT2Tokenizer[10] [19] [17]. Through this series of careful experimental steps, we aimed to explore the model's ability to parse and understand image content, and to generate accurate descriptive text based on that content.

We then turn our attention to experiments with the lyrics generation model, which evaluates the quality of rhyming between the generated text and the target label by adding the computation of rhyming loss to the standard language model loss through the customized GPT2RhymeModel class. The design of this dual loss function not only highlights the innovative point of our study, but also aims to facilitate the model's performance in generating semantically coherent texts that follow rhyming patterns. Through this two-part test and evaluation, we demonstrate the potential of the model's application and its effectiveness in two different domains: image description generation and lyrics composition, proving the effectiveness of our approach and the flexibility of the model.

3.1 Image Description Model Examples

In the results section of our project, a series of tests were implemented in order to fully evaluate the performance of the developed visual-linguistic model for automatic image description generation. These tests are based on the Python programming language and utilize the widely used COCO 2017 validation dataset provided by TensorFlow Datasets (TFDS), as well as the VisionEncoderDecoderModel developed by Huggingface, ViTFeatureExtractor, and GPT2Tokenizer [10] [19] [17]. The COCO dataset is a large-scale dataset widely used in the field of computer vision, characterized by its large diversity and exhaustive annotation. Specifically, it contains over 200,000 images covering a wide range of scenes and backgrounds, from busy city streets

to magnificent outdoor landscapes. Each image is equipped with fine-grained annotation information, such as object bounding boxes, segmentation masks, and key point localization, providing powerful support for accurately identifying and locating objects in images. In addition, the COCO dataset defines 80 different object categories that cover common objects in human daily life, including people, vehicles, furniture, food, and animals, etc. This broad category coverage makes it an indispensable resource in machine learning tasks [17].

First, we loaded the pre-trained VisionEncoderDecoder model, which is specifically designed for image description generation tasks, combining ViT (Visual Transformer) as an encoder, and GPT-2 as a decoder. The model and the associated feature extractor are loaded via the from_pretrained method, which ensures that we can directly utilize pre-trained weights and parameters on large-scale datasets. Also, GPT2Tokenizer was used to process the generated text sequences.

For preparing the test data, we loaded the validated segmentation part of the COCO dataset from TFDS, which contains detailed annotation information. For the testing process, we defined the prepare_image function, which takes an image input and converts it into a format that can be processed by the model through a feature extractor, and subsequently the model generates descriptive text based on these processed features.

To demonstrate the testing process, we iterated over a single sample from the dataset, which was primarily for demonstration purposes. During the iteration process, we converted the TensorFlow image tensor into a PIL image object so that it can be processed correctly by the feature extractor. Subsequently, we call the prepare_image function to generate a description and print the results via standard output. One example is presented in Figure 3.1, and more examples are at Appendix A.

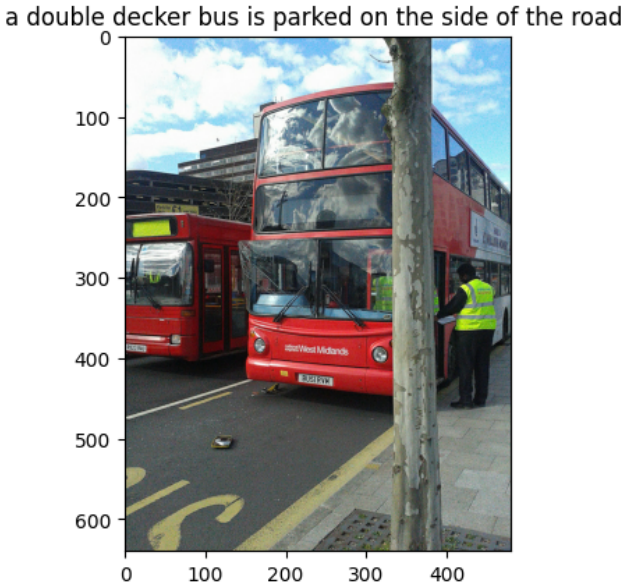


Figure 3.1: Image and caption generated by VisionEncoderDecoder model

3.2 Lyrics Generation Model Examples

We trained and fine-tuned the model using GPT2Tokenizer and GPT2LMHeadModel, as well as the GPT2RhymeModel class customized specifically for this task. The customized model not only computes the standard language model loss, but also additionally computes a rhyme loss for evaluating the quality of rhyming between the generated text and the target tag. This unique design of the dual loss function is an innovative point of our research, aiming to facilitate the model's performance in generating semantically coherent and rhyme-compliant texts. After training, we started lyrics generation. For this purpose, we employed the Python programming language, combined with a powerful text generation pipeline provided by the Transformers library, to implement and test our model [16]. First, we designed an interface that, through the use of the pickle library, allowed us to directly load a model that had been specifically trained, such as a Drake-style model [23]. Immediately after that, we extracted a tokenizer from the gpt2 pre-trained model using the GPT2Tokenizer, which is used to process the input text and convert it into a format that the model can understand. In the result generation phase, we employed a text generation pipeline that specified our trained model and its prompt as input. In order to assess the model's creativity and flexibility, we provided a specific prompt, such as "a teddy bear sitting on a shelf next to a cup". The purpose of this prompt is to explore the model's ability to generate lyrics when receiving a prompt from an image description model. After setting the maximum generation length to 70 words, the number of sequences returned to 1, and adjusting the temperature parameter to 0.8 to control the innovativeness of the generated text, we observed the model's output. Figure 3.2 shows the lyrics model generated using "a teddy bear sitting on a shelf next to a cup" as input for the simulated image description. More lyrics examples are at Appendix B.

A teddy bear sitting on a shelf next to a cup of coffee,
In the quiet, thoughts start to race, never sloppy.
In this moment, life's bitter and sweet, like toffee,
Yeah, it's a show, sitting row by row, quite lofty.
Had a thought, in my mind, it just wouldn't back down,
Life's a circus, sometimes you're the clown.
On the edge, where dreams and reality drown,
In this beat, let's lose the frown, wear the crown.

Figure 3.2: Example of generated lyrics based on a given prompt.

Chapter 4

DISCUSSION

In this section we will focus on evaluating the quality of our model for image description and lyrics generation.

4.1 Image Description Evaluation

In order to accurately evaluate the performance of our VisionEncoderDecoder model in the task of automatically generating image descriptions, we executed a series of rigorous quantitative tests. These tests were performed on the validation set of the widely recognized COCO 2014 dataset provided by TensorFlow Datasets, with the top 20 samples from the dataset selected for in-depth analysis [17]. This subset was selected to balance the computational efficiency of the evaluation process with the representativeness of the results. Each image is transformed by the feature extractor into a format that can be processed by the model, which subsequently outputs descriptive text based on these features.

We compared the descriptions generated by the model with actual image descriptions, which served as evaluation criteria for the annotation information from the COCO dataset. To perform this comparison, we created two dictionaries, one recording the actual captions and the other recording the captions generated by the model for the corresponding images.After normalization using the PTBTokenizer tool, we employed the CIDEr (Consensus-based Image Description Evaluation) scoring mechanism to evaluate the quality of the model-generated descriptions. The CIDEr metric is a widely adopted standard in the field of automatic image description generation to evaluate the quality of descriptions. It achieves this by calculating the similarity between a candidate description and a set of reference descriptions, with special emphasis on TF-IDF weighting of the vocabulary and n-gram consensus [26]. CIDEr is designed to take into account the importance of informative vocabulary in descriptions, and to capture semantic coherence and description fluency through n-gram analysis [26]. In addition, CIDEr guarantees the fairness of scoring through normalization, which provides reliable evaluations even when the number of reference descriptions varies [26]. This consistency scoring method ensures the ability of candidate descriptions to reach consensus with multiple reference descriptions in reflecting key features of the image, which in turn measures the accuracy and

quality of the descriptions. Therefore, CIDEr is used as a central evaluation tool in academic research and various challenges to comprehensively assess the semantic consistency of model-generated image descriptions with those given by human annotators. Our study employs the CIDEr metric to evaluate the quality of model-generated descriptions, which provides us with key indicators of the model's performance in image understanding and language generation.

In addition, we used a comprehensive VisionEncoderDecoderModel to generate image descriptions and compared its performance to several benchmark models. These benchmark models include Show and Tell, Show, Attend and Tell, and Up-Down (Bottom-Up and Top-Down Attention) models [28] [27] [1]. Show and Tell, Show, Attend and Tell, and Up-Down are seminal models in the field of image captioning, a task that combines computer vision and natural language processing to generate descriptive text for images.

Show and Tell, proposed by Vinyals et al, was one of the first models to effectively utilize neural network architectures for image caption processing [27]. It uses a convolutional neural network (CNN) to encode the visual features of an image, and then uses a long short-term memory (LSTM) network to decode these features into coherent sentences. The model shows that such architectures can be trained end-to-end to generate high-quality subtitles, thus marking an important step forward in the field [27].

Show, Attend and Tell is added with an attention mechanism. This innovative technique, proposed by Xu et al [28]. It allows the model to dynamically focus on different parts of the image as each word of the caption is generated. The attention mechanism allows the model to create more detailed and contextually relevant descriptions by highlighting specific areas of the image that are relevant to the generated text.

The "Top-Bottom (Bottom-Up and Top-Down Attention)" approach proposed by Anderson et al [1]. It further refines the attention-based approach by using two separate attention mechanisms. The bottom-up mechanism utilizes object detection to propose salient image regions, while the top-down mechanism selectively focuses attention on these regions using the context generated so far. This combination allows for more accurate and context-aware captioning, as the model can strike a balance between focusing on specific objects and overall scene context.

Each of these models contributes to the development of more sophisticated and efficient image captioning systems, setting new standards for the accuracy and detail of automatically generated natural language descriptions of visual content.

All models were evaluated on the widely used COCO dataset on the CIDEr score, which quantifies the quality of the model-generated descriptions and the consistency between the reference descriptions. In this comparison, the VisionEncoderDecoderModel demonstrates excellent performance with a CIDEr score of 2.462, which is significantly higher than the scores of the other models on the same task. Specifically, the Show and Tell model scores about 1.0, the Show, Attend and Tell model scores vary between 1.0 and 1.1, and the Up-Down model scores about 1.2.These results suggest that although previous models have achieved respectable results on image description tasks, our proposed VisionEncoderDecoderModel sets new standards in capturing image content and generating high-quality natural language descriptions.

Table 4.1 shows the results of evaluation.

Model Name	CIDEr Score
VisionEncoderDecoderModel	≥ 2.0
Show and Tell	1.0
Show, Attend and Tell	1.0–1.1
Up-Down (Bottom-Up and Top-Down Attention)	1.2

Table 4.1: Comparison of CIDEr Scores across different models on the COCO dataset.

In summary, VisionEncoderDecoderModel outperforms existing state-of-the-art models in image description generation tasks. This result highlights the sophistication and effectiveness of our model in handling vision-linguistic tasks, especially in understanding complex image content and generating rich and accurate descriptions. Moreover, the success of the model lays the foundation for potential future improvements and innovations in a wider range of visual-verbal fusion applications.

4.2 Lyrics Evaluation

4.2.1 Objective Evaluation

In conducting the objectivity assessment, this project adopts a specially designed function to compute the performance of the generated lyrics and their language models on the dimensions of Rhyme Density (RD) and Perplexity (PPL). The concept of rhyme density is derived from the work of Malmi et al [13]. This metric effectively reflects the fluency of rhyming lyrics and the weight of their rhyming components [13]. Given that rhyming elements may be distributed within or at the end of lines in hip-hop music lyric writing, this study quantifies rhyme density by assessing the percentage of rhyming words in the full text of the lyrics. Meanwhile, perplexity, as a measure of the accuracy of probabilistic models on sample prediction, can reflect the level of uncertainty of language models in predicting new words based on the above. In general, a higher perplexity metric implies that the language model is more uncertain in predicting new words based on the context, and thus, we expect the model obtained through training to have a lower perplexity value. Given that the GPT-2 model itself does not have the property of measuring complexity directly, this study calculates model complexity indirectly by introducing a specialized function based on model loss. Ideally, models with lower complexity would be able to predict the data in the test set more accurately, showing higher predictive certainty.

Our objective evaluation results are summarized in 4.2. The table presents the results of an objective evaluation of the rhyme density (RD) and perplexity (PPL) of lyrics from different sources - including those generated by models trained on Drake, ASAP Rocky, and Eminem - as well as their original lyrics. Rhyme Density serves as a measure of rhyme frequency, with higher values indicating more frequent use of rhymes in lyrics. We selected a sample of 10 songs under each style and calculating its average metric respectively.

In this evaluation, the lyrics generated by the Drake-trained model showed the highest average

rhyme density (0.3933), exceeding the rhyme density of Drake's original lyrics (0.2614). Similarly, the models trained by ASAP Rocky and Eminem also produced higher rhyme densities than their original lyrics. In the perplexity assessment, this metric quantifies the uncertainty of the language model in predicting the next word given the above. On this dimension, the GPT-2 pre-trained model exhibits the lowest perplexity (162.47), which suggests that it has higher accuracy and certainty in predicting new words. In contrast, the model trained by Eminem showed the highest degree of perplexity (815.46), indicating greater uncertainty in its predictions.

In conclusion, we conducted a meticulous rhyme density (RD) analysis of songs of various styles, and selected a sample of 10 representative songs under each style for the average calculation. We found that the model-generated lyrics not only equal the original lyrics in terms of rhyme density, but even achieve a slight improvement in some cases, which fully demonstrates the model's excellent ability in grasping rhyme structure. This result reflects the significant progress made by the model in learning and simulating rap artists' rhyming techniques, especially in imitating their unique rhythmic and rhyme patterns.

However, an in-depth analysis of the lyrical content generated by the model reveals some thought-provoking phenomena. In the pursuit of rhythmic precision, some of the lyric fragments showed an undue sacrifice of semantic richness, as well as repetitive stanzas, which may have led to an artificial increase in the model's rhyming density scores. In particular, the model's performance exposed some limitations in terms of semantic diversity and depth. These problems may be due to over-emphasizing the learning of rhyming patterns in the design of the loss function for model training, without properly balancing the importance of semantic coherence and innovation.

In addition, the results of the perplexity level (PPL) analysis of the model-generated lyrics showed a large gap in the certainty of linguistic predictions of the models trained by specific artists compared to the low perplexity level of the pre-trained models of the GPT-2. This implies that although the models were successful in modeling rhyming patterns of specific art styles, there is still room for improvement in the accuracy of predicting contextual neologisms. The pre-trained models have a strong generalization ability, which is particularly important when dealing with complex and diverse hip-hop lyrics, whereas the models trained for specific artist styles need to further improve the accuracy and flexibility of their linguistic predictions in future studies.

	Average RD	**PPL**
Drake (Trained)	0.3933	326.86
Drake (Original)	0.2614	(None)
ASAP Rocky (Trained)	0.236	401.45
ASAP Rocky (Original)	0.2499	(None)
Eminem (Trained)	0.2808	815.46
Eminem (Original)	0.2674	(None)
GPT-2 Pre-trained (Not Fine-tuned)	(None)	162.47

Table 4.2: Rhyme Density (RD) and Perplexity (PPL) for generated and original lyrics

4.2.2 Subjective Evaluation

In the process of careful subjective assessment of the generated lyrics, this study collected feedback and evaluations from the target audience groups through a well-designed questionnaire. The questionnaire was designed to assess the quality of the generated lyrics in terms of three key dimensions-logical plausibility, rhythmic expressiveness, and stylistic fidelity. A total of 33 individuals with a base of knowledge about hip-hop culture participated in the survey, giving detailed scores from 0 (worst) to 10 (best) to the provided sample of lyrics.

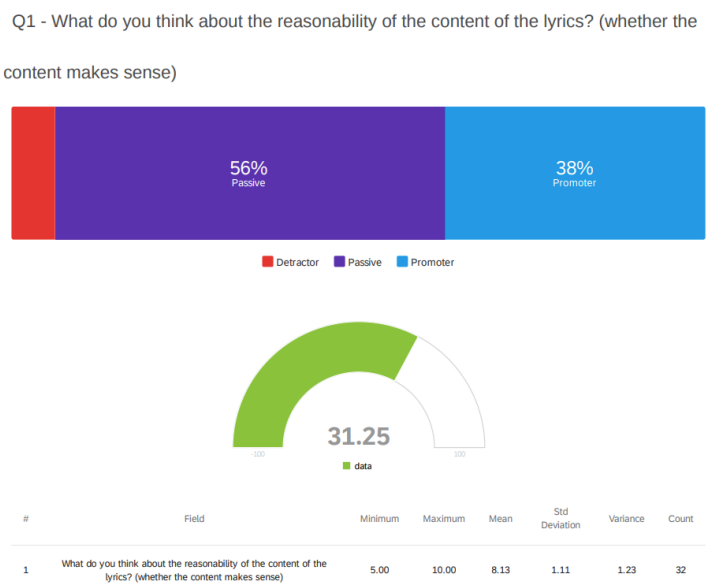
A summary of the survey results shows that participants gave overall positive feedback on the model-generated lyrics. In terms of logical soundness, the average score reached 8.13, reflecting that participants generally recognized the logical coherence of the lyrics' content. Rhyming expressiveness was also rated quite favorably, with a mean score of 7.56, indicating that participants were satisfied with the rhyming technique of the lyrics. However, the mean score for stylistic fidelity was 6.78, a relatively low score pointing to room for improvement in stylistic capture.

Figure 4.1, 4.2 and 4.3 provide a visual presentation of participants' subjective evaluations of the generated lyrics, which contains the distribution of feedback for the three questions. Responses to each question were categorized into three groups: critics (Detractor), neutrals (Passive), and promoters (Promoter).

In the first question, participants evaluated the plausibility of generating lyrical content, i.e., whether the content was meaningful. The results showed that 38% of the participants were Promoters, who considered the lyrics content to be reasonable; in contrast, 56% of the participants were neutral, compared to 6% of the critics. The second question focused on the rhyme quality of the generated lyrics. In this regard, 28% of the participants were promoters who thought the lyrics had good rhymes; while half of the participants were neutral and 22% were critics. The final question asked participants what they thought about generating lyrics that captured the style of a particular rapper. In response, 16% of the participants gave positive feedback, 47% were neutral, and critics made up 38%.

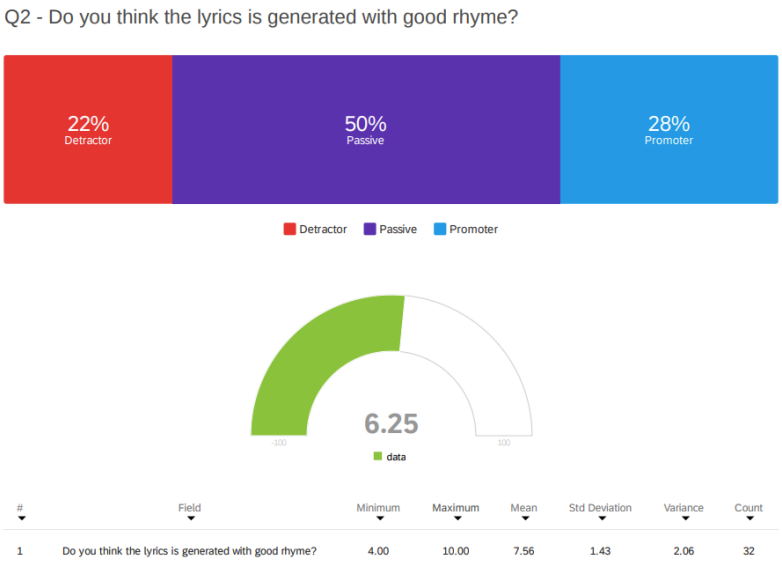
We observe that despite the wide distribution of attitudes, a high percentage of the participants had reservations regarding style evaluation, implying that accuracy of style reproduction is an important direction for future research and model optimization.

Taken together, these feedbacks show that the evaluators were generally satisfied with the model's performance in terms of rhyme and content plausibility, with the performance in rhyme in particular being more prominent. This reflects, in part, the effectiveness of the GPT-2 model with a multiple attention mechanism and a specifically designed loss function, as well as the efficacy of the post-processing step in enhancing the rhythmicity of the lyrics. However, the post-processing may have affected the coherence and overall consistency of the lyrical content while enhancing the rhythm, which may be a potential reason for the slightly lower plausibility scores than the rhyme scores. As for the relatively low satisfaction with style capture, this may point to the limitations of our model training method in capturing specific artist styles. We relied solely on data from singers of different styles for model training without



#	Field	Minimum	Maximum	Mean	Std Deviation	Variance	Count
1	What do you think about the reasonability of the content of the lyrics? (whether the content makes sense)	5.00	10.00	8.13	1.11	1.23	32

Figure 4.1: Survey results Q1



#	Field	Minimum	Maximum	Mean	Std Deviation	Variance	Count
1	Do you think the lyrics is generated with good rhyme?	4.00	10.00	7.56	1.43	2.06	32

Figure 4.2: Survey results Q2

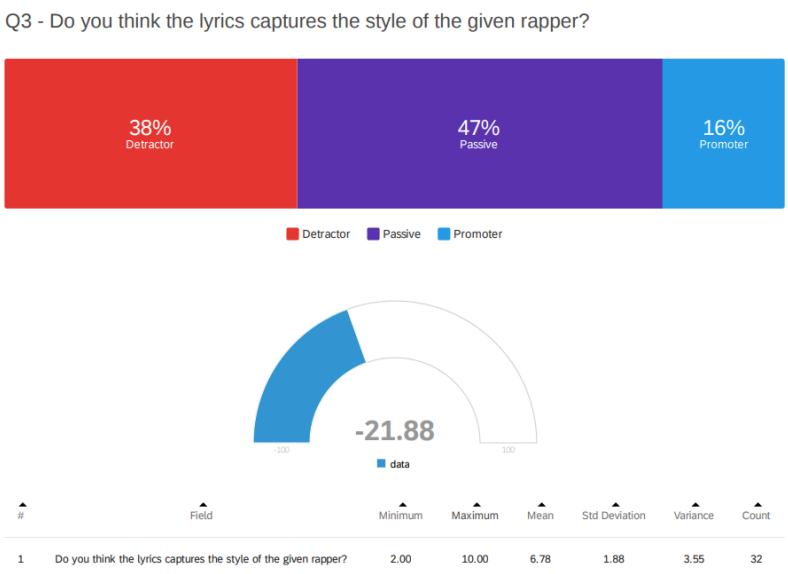


Figure 4.3: Survey results Q3

further optimizing the model to capture and reproduce each artist's unique creative characteristics in greater detail. This finding suggests that in our future work, we need to explore more in-depth and refined model training and optimization strategies to more accurately reproduce and approximate the stylistic characteristics of each artist.

4.3 Future Work

In terms of future work, we plan to further deepen and extend the exploration of this research in the automatic generation of hip-hop lyrics. This project has preliminarily achieved the design goal and successfully developed a stylistic hip-hop lyrics generator that creates lyrics with rationality and rhyming according to a image. In our future research, we will work in the following directions.

First, we will deepen the loss function. We intend to introduce more complex and fine-grained loss functions to capture and learn more dimensional features in lyrics, e.g., focusing on style capture. Currently, the loss function focuses on rhyming and regular semantic dimensions, while the calculation of rhyming loss is relatively simple. In the future, we will explore more fine-grained rhyme loss computation methods and try to incorporate other music theory and linguistic elements to achieve more comprehensive quality control of lyrics.

Second, we hope to expand the lyrics dataset. We plan to enrich the training base of the model by collecting more diverse lyrics data of hip-hop music. By introducing more rap artists' works, lyrics from different regions and cultural backgrounds, we can further improve the model's ability to understand and generate diverse styles, and thus optimize the quality of the generated lyrics.

Third, we plan to improve the parameter optimization strategy. The current model parameters, especially the setting of weights in the loss function, mainly rely on manual adjustment.

In the future, we plan to use more advanced automation techniques, such as hyper-parameter optimization algorithms in machine learning, to determine these weights, thus reducing human intervention and improving model performance.

Fourth, we would like to realize the intelligence of the post-processing mechanism: the current post-processing step of the model requires human intervention, which is time-consuming and may interfere with the accuracy of the original generated results. Therefore, we hope to develop a more intelligent post-processing system that uses deep learning techniques to automatically judge and perform lyric corrections to maintain semantic coherence and rhyming accuracy, while minimizing human intervention.

Finally, we would like to achieve more multi-modal feature fusion. We will explore incorporating more modalities into the model, such as audio and video information, which will help the model provide richer contextual information when describing images and write more creative lyrics.

With the above measures, we believe that the future system will take a more solid step in improving the efficiency and expanding the creative possibilities of hip-hop lyrics creation, and open up new paths for the application of deep learning in the music field.

Chapter 5

CONCLUSIONS

In this study, an automatic hip-hop lyrics generation system that combines image recognition and natural language processing capabilities is successfully developed by integrating cutting-edge machine learning and deep learning techniques. The system is not only capable of generating descriptive text based on the content of a given image, but also of creating lyrics that match a specific hip-hop style, demonstrating the high potential of the model for multimodal learning and idea generation. Our model, based on VisionEncoderDecoderModel and GPT-2, is able to recognize image content and then create hip-hop lyrics that match the style of a specific artist. We employ a pre-trained VisionEncoder-DecoderModel for image description and incorporate a fine-tuned language model based on the Transformer architecture to generate creative lyrics. First, we employed a pre-trained VisionEncoderDecoder model specifically designed for the task of generating image descriptions, combining the Vision Transformer (ViT) as an encoder and the GPT-2 as a decoder. The model and the associated feature extractor are loaded by a pre-training method, which ensures that we can directly utilize the pre-trained weights and parameters on the large-scale dataset. The model can pass descriptions of images as input to the lyrics generation model for lyrics generation. Next, we trained and fine-tuned the lyrics generation model using GPT2Tokenizer and GPT2LMHeadModel as well as the GPT2RhymeModel class customized specifically for this task. The customized model not only computes the standard language model loss, but also additionally computes the rhyme loss to evaluate the quality of the rhyme between the generated text and the target label. This uniquely designed dual loss function is an innovative point of our research and aims to facilitate the model's performance in generating semantically coherent and rhyme-compliant text. The lyrics generation model can create lyrics based on the descriptions delivered by the image description model according to the style of a particular singer. Such an approach realizes our whole process from image to lyrics creation. In the model evaluation section, for our VisionEncoderDecoderModel, we took an objective evaluation approach to measure its performance. We used the CIDEr (Consensus-based Image Description Evaluation) metric and compared VisionEncoderDecoderModel with other peer models. For our lyrics generation model, given that the assessment of lyrics quality usually involves subjective tastes and emotional resonance, we comprehensively evaluated the lyrics generation model on both objective and sub-

jective dimensions. For the objective assessment, we measured the quality of the generated lyrics based on metrics such as rhyme density and perplexity. For the subjective assessment, we conducted a human-participant survey in which participants evaluated the generated lyrics based on multiple dimensions such as fluency, creativity, emotional expression, and stylistic mimicry. In the future, we will improve the accuracy of the model and the quality of the lyrics through innovative approaches and explore more creative possibilities.

REFERENCES

[1] Peter Anderson et al. "Bottom-Up and Top-Down Attention for Image Captioning and Visual Question Answering". In: *arXiv preprint arXiv:1707.07998* (2018).

[2] Y. Bai et al. *Are Transformers More Robust Than CNNs?* arXiv. Nov. 2021. DOI: 10.48550/arXiv.2111.05464.

[3] *Beautiful Soup Documentation —Beautiful Soup 4.12.0 documentation*. https://www.crummy.com/software/BeautifulSoup/bs4/doc/. Accessed: April 21, 2023.

[4] Nell Shaw Cohen. *Music Inspired by Visual Art*. Accessed: February 24, 2024. Nov. 2015.

[5] emakpati. *How to Collect Song Lyrics with Python*. https://towardsdatascience.com/song-lyrics-genius-api-dcc2819c29. Accessed: May 18, 2023. Dec. 2020.

[6] Hugging Face. *ViT - transformers*. Accessed: 2024-02-27. 2024.

[7] *Genius | Song Lyrics & Knowledge*. https://genius.com/. Accessed: April 7, 2023.

[8] The Guardian. *Hip-hop's iconic images and the stories behind them in pictures*. Accessed: February 24, 2024. 2018.

[9] M. Hernandez et al. *Automatic Generation of Hip-Hop and Rap Lyrics*.

[10] Hugging Face. *Vision Encoder Decoder Model Documentation*. https://huggingface.co/docs/transformers/model_doc/vision-encoder-decoder. Accessed: 2023-02-22. 2023.

[11] F. B. Krohn and F. L. Suazo. "Contemporary Urban Music: Controversial Messages in Hip-Hop and Rap Lyrics". In: *ETC Rev. Gen. Semant.* 52.2 (1995), pp. 139–154.

[12] 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*. Accessed: May 18, 2023. Association for Computational Linguistics. Santa Fe, New Mexico, USA, 2018, pp. 641–652.

[13] Eric Malmi et al. *DopeLearning*. https://dl.acm.org/doi/10.1145/2939672.2939679. Accessed: May 18, 2023.

[14] J. Maurício, I. Domingues, and J. Bernardino. "Comparing Vision Transformers and Convolutional Neural Networks for Image Classification: A Literature Review". In: *Appl. Sci.* 13.9 (Jan. 2023). DOI: 10.3390/app13095521.

[15] J. Maurício, I. Domingues, and J. Bernardino. "Comparing Vision Transformers and Convolutional Neural Networks for Image Classification: A Literature Review". In: *Applied Sciences* 13.9 (Jan. 2023). DOI: 10.3390/app13095521.

[16] *Models - Hugging Face*. https://huggingface.co/models. Accessed: April 7, 2023.

[17] *MS COCO: Common Objects in Context*. https://cocodataset.org/#home. Accessed: 2024-03-05.

[18] Z. Obermeyer and E. J. Emanuel. "Predicting the Future —Big Data, Machine Learning, and Clinical Medicine". In: *N. Engl. J. Med.* 375.13 (Sept. 2016), pp. 1216–1219. DOI: 10.1056/NEJMp1606181.

[19] *OpenAI GPT2*. https://huggingface.co/docs/transformers/model_doc/gpt2. Accessed: May 11, 2023.

[20] A. Parrish. *pronouncing: A simple interface for the CMU pronouncing dictionary*. https://github.com/aparrish/pronouncingpy. Accessed: April 21, 2023.

[21] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. *GloVe: Global Vectors for Word Representation*. https://nlp.stanford.edu/projects/glove/. Accessed: April 21, 2023.

[22] Peter Potash, Alexey Romanov, and Anna Rumshisky. *GhostWriter: Using an LSTM for Automatic Rap Lyric Generation*. https://aclanthology.org/D15-1221/. Accessed: May 18, 2023.

[23] Python Software Foundation. *pickle —Python object serialization*. https://docs.python.org/3/library/pickle.html. Accessed: 2023-09-28. 2023.

[24] R. Rehurek. *gensim: Python framework for fast Vector Space Modelling*. http://radimrehurek.com/gensim. Accessed: April 21, 2023.

[25] A. Vaswani et al. "Attention is All you Need". In: *Advances in Neural Information Processing Systems*. Accessed: May 18, 2023. Curran Associates, Inc., 2017.

[26] Ramakrishna Vedantam, C. Lawrence Zitnick, and Devi Parikh. "CIDEr: Consensus-based Image Description Evaluation". In: *arXiv preprint arXiv:1411.5726* (2015). To appear in CVPR 2015. DOI: 10.48550/arXiv.1411.5726. arXiv: 1411.5726 [cs.CV].

[27] Oriol Vinyals et al. "Show and Tell: A Neural Image Caption Generator". In: *arXiv preprint arXiv:1411.4555* (2015).

[28] Kelvin Xu et al. "Show, Attend and Tell: Neural Image Caption Generation with Visual Attention". In: *arXiv preprint arXiv:1502.03044* (2015).

[29] L. Xue et al. *DeepRapper: Neural Rap Generation with Rhyme and Rhythm Modeling*. arXiv. July 2021. DOI: 10.48550/arXiv.2107.01875.

APPENDIX A

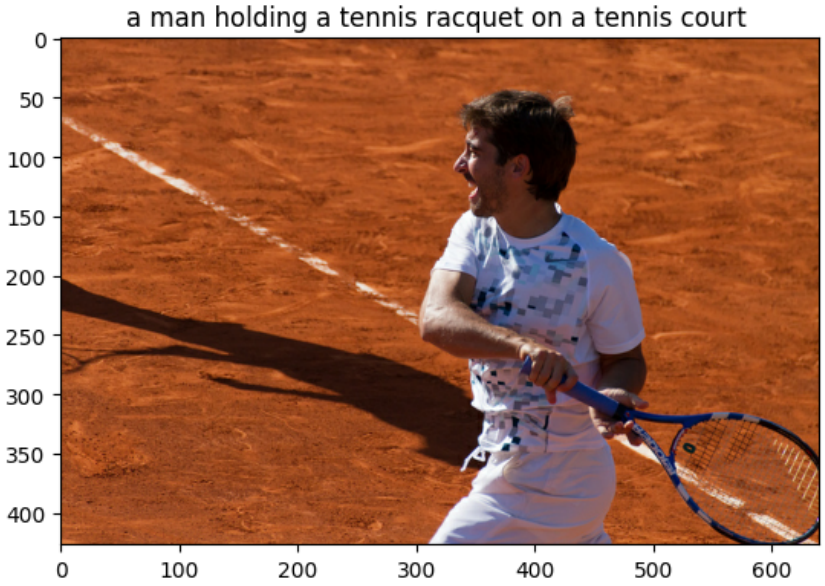


Figure A.1: Image and caption generated by VisionEncoderDecoder model example 1

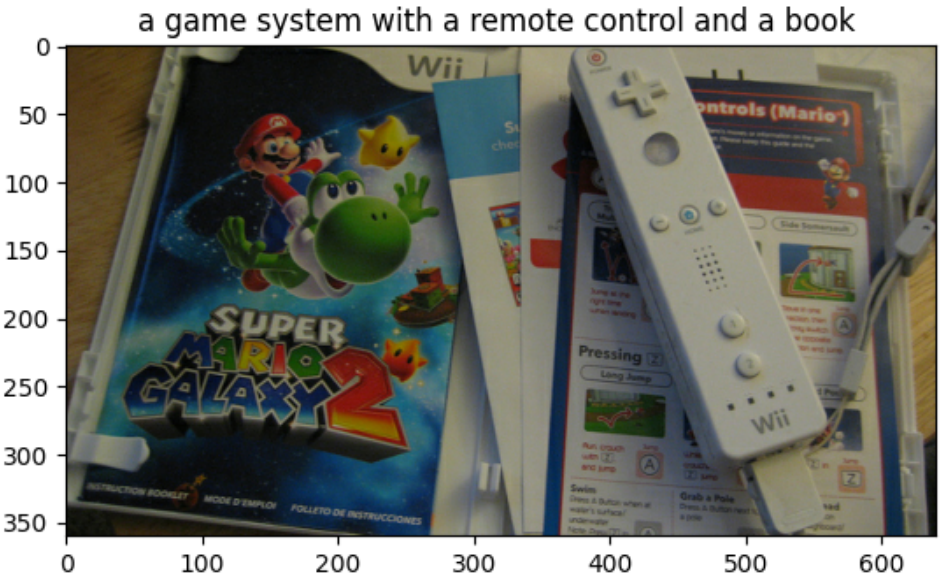


Figure A.2: Image and caption generated by VisionEncoderDecoder model example 2

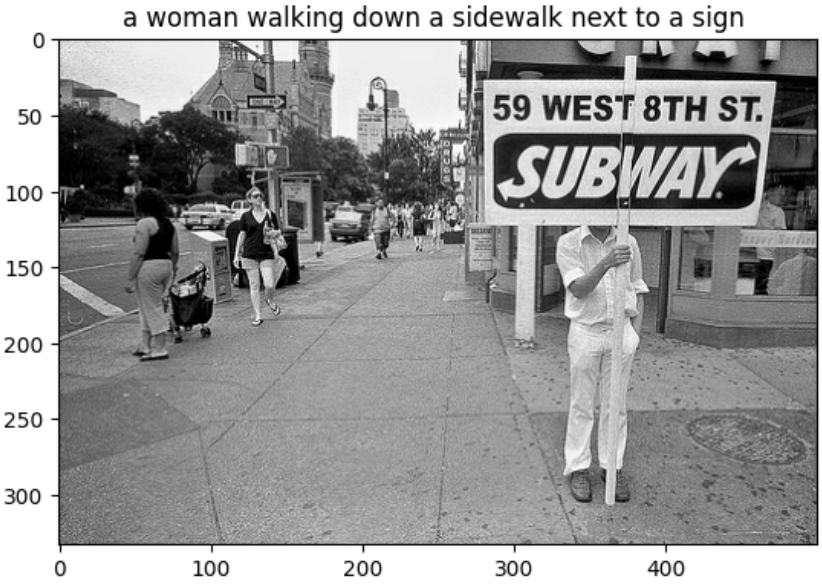


Figure A.3: Image and caption generated by VisionEncoderDecoder model example 3

Figure A.4: Image and caption generated by VisionEncoderDecoder model example 4

Figure A.5: Image and caption generated by VisionEncoderDecoder model example 5

APPENDIX B

(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