Deep-Hop Rapper: Using LSTM and Transformer for Rap Lyric Generation

Stanford CS224N {Custom, Default} Project

Aaron Han
Department of Computer Science
Stanford University
than21@stanford.edu

Roy Park
Department of Computer Science
Stanford University
rpark3@stanford.edu

Abstract

Natural language generation (NLG) techniques have developed substantially to a point where automatically generated languages are basically indistinguishable from human-made counterparts. To think that they would be able to aid in creative fields such as news article writing, coding, and poetry was almost unthinkable previously. Despite NLG's advancements and incorporation of the technique in these creative fields, it has yet to reach a prominent space in creative writing: hip-hop, or rap, lyric despite the genre's rise to the mainstream in the past two decades. In this project, we test and compare the effectiveness of two language generation models for hip-hop: LSTM and Transformer. The theory behind these two models suggest that Transformers, even without pre-training on the English language, should perform better than LSTM through the use of attention scores. Our quantitative evaluation of the two techniques show that there are marginal differences in performance between LSTM and Transformer-based models. However, a closer qualitative examination shows that the Transformer-based model indeed performs better in terms of grammatical and contextual coherence of the generated lyrics.

1 Key Information to include

- Mentor: Manan Rai
- External Collaborators (if you have any): No
- Sharing project: No

2 Introduction

For the past few decades, hip-hop has dominated top musical charts in both the US and the rest of the world. Its social implications, cultural significance, lyricism, dynamism, and charisma are like no other genres. Youths are highly influenced by the culture of hip-hop. One of the most essential elements of hip-hop is lyrics. In the art of writing rap lyrics, technical, phonic, and connotative factors determine the quality the lyrics. The technical factor of rap lyrics include punchlines and rhymes. The phonic factor include rhythm, flow, and euphony. Connotative factors include content, story-telling, social implications, and authenticity.

Natural language generation techniques have developed substantially to a point where their products are indistinguishable from human-made counterparts. It can effectively provide assistance in various creative fields such as news article writing, coding, even poetry. However, these techniques have not been used to much success in hip-hop lyric generation. Other academic papers use outdated techniques or depend on pre-existing lyrics (not 100% original generation).

We applied these techniques to hip-hop lyric generations and compared results between LSTM and Transformer models in this project.

The key idea of our approach is to assess the significance of attention in language models based on both quantitative and qualitative metrics to evaluate how well our lyrics generated by our models display the previously-mentioned three main factors of rap lyricism.

3 Related Work

Hip-hop lyric generation has not been seriously attempted with the more modern NLG techniques that we are planning on using. The paper that shares the most similar goals as ours implemented a model that generates lyrics by using existing bars that match rhymes with the preceding bar.[1] This method has limitations in the coherence of the entire lyric as rhymes are the only determining factor for the lyrics, and we will be attempting to improve the performance of rap lyric generators in regards to both coherence as well as the stylistic closeness with a given rapper.

However, LSTM models have been used to success in a genre very similar to hip-hop: sonnets. Augmenting a simple LSTM model with separately trained rhyme and pantameter models have shown great success in generating sonnets extremely similar to Shakespeare to the point where human evaluators had 50% accuracy in determining whether a sonnet has been writen by Shakespeare or AI. [2] While the objectives of this paper are a little different from ours, it still indicates a positive outlook for the possibility of generating rap lyrics that closely mimic a rapper's style.

4 Approach

Note that we wish to both explore rap lyric generation of language models and also assess the effectiveness of attention scores in language generation. In order to analyze the advantage of attention scores, we decided to implement and train LSTM model and Transformer model. We specifically chose LSTM because it is one of high-performing neural network models that do not utilize attention scores. We used a single-layered many-to-one LSTM model. We also chose to use multi-head attention for our Transformer model to leverage the power of attention. We also used dropout for our encoder in order to prevent overfitting. For both models, we use exponential learning rate decay to help speed up our training. The implementation of Transformer model was inspired by a tutorial from Pytorch.org [3].

We decided to train our models on the lyrics data of the two following artists: Eminem and Kanye West. We have chose these specific artists due to the abundance of lyrical data and due to their distinct style of language in terms of their lyrics. Eminem is a renowned lyricist who once mentioned he often reads dictionaries to acquire expansive vocabulary and linguistic skills. On the other hand, Kanye West is a producer-turned-rapper who is well-known for his innovative musical production rather than his lyricism.

We implemented an LSTM and Transformer model in PyTorch. For our LSTM model, we used a single-layered LSTM model class from PyTorch. Specifically, we used a many-to-one LSTM. We wanted our model to take one or more words as input to generate each word, which includes the newline character "\n".

We also decided not to use pre-trained Transformer for "fairer" comparison. since pre-trained Transformer models would perform much better in generating coherent sentences and can be just fine tuned for rhyme scheme and rap flow

Since we wanted to explore the capability of language models to generate rap lyrics in the style of a specific artist, we decided to train these models with text data of lyrics of tracks by Eminem and Kanye West. Specifically, we trained a total of four different model in which each one of LSTM and Transformer model is trained on Eminem and Kanye West separately, expecting each model to pick up specific style of language.

5 Experiments

5.1 Data

The data for the models were retrieved from Genius.com, a hip-hop lyrics community and database. [4] Lyrics for every single song that Genius.com detects as Eminem and Kanye West's songs were

retrieved from the Genius.com API in .json format. The .json files from the API call contained irrelevant information, which was purged while the lyrics were pulled into .txt files that was formatted line-by-line to follow Hip-Hop's "bar" structure, a key component to the rhyming structure and flow of the lyrics. The lyrics also contained headers irrelevant for the purposes of our training data such as "[Verse 1]" or "Godzilla Lyrics," which were eliminated through regular expression functions. The dataset included transcripts and translated lyrics, and since our purposes were to mimic Emine and Kanye West's lyrics in English in particular, they were eliminated as well.

The input to the model was a line of text between 4 to 10 words. We used iconic lines from Eminem and Kanye West such as "His palms are sweaty knees weak arms are heavy" from Eminem's *Godzilla* and "No one man should have all that power" from Kanye West's *Power*. This was to achieve the following:

- Generate lyrics based on a starting line or a topic – we intended this behavior as the ability to generate lyrics to the style of an artist given a word, topic, or a starting line. This also prevents errors since the models will have been trained on these words. We can also check whether the model is overfitted this way, since the generated lyrics would be extremely similar to the song that the starting line is from if the models are overfitted.
- Use BLEU scores as an evaluation metric. While we do not want the models to completely, we do want to measure how well the models mimic the original lyrics, and as a result, the lyric style of an artist. Generating the lyrics based on a line from existing songs allow us to use BLEU as a metric.

We also used a smaller dataset for Transformer models than the LSTM models. The entire dataset included some headers that could not be purged. While the LSTM models were not heavily affected by the existence of lower quality data, Transformer models noticeably struggled with the existence of these headers. As such, we used 25% of the dataset that we were absolutely confident in the quality of for the Transformer models. Using smaller datasets for Transformer models not only reduced training times to similar levels as training LSTM models but also led to much quicker convergence of losses.

5.2 Evaluation method

We used two quantitative metrics. First is rhyme density, which measures the frequency of rhymes in text data. For reference, most respected lyricists in hip-hop have scored at an average of 1 or higher in this metric. [1]

However, there are clear limitations to the quantitative metrics we are using. While rhyme density is very good at measuring very obvious rhymes e.g. power and tower, it struggles with the fact that rappers may deviate from standard pronunciations . For instance, Eminem talks about how "door hinge" can be rhymed with "orange," a word infamous for not having rhyming words, as long as you alter In fact, Eminem scored lower in rhyme density than Nicki Minaj, who is not known to be a highly touted lyricist due to this reason. [1] BLEU is limited for our purposes as while we are mimicking existing lyrics to an extent, that is not our ultimate goal we would like to generate original lyrics. BLEU only evaluates how close the generated text is to the reference text, which causes the limitations.

As such, we focused more of our evaluation on the qualitative results. Because we were not using a pre-trained Transformer model, we focused on whether the models can generate both grammatically and contextually coherent lyrics.

5.3 Baseline

The baseline is lyrics generated from `https://deepbeat.org/`. Instead of using completely newly generated lyrics like our model, the model behind this website uses preexisting lyrics that rhyme to generate lyrics (e.g. if the user input is "something wrong I hold my head," from *All of the Lights* by Ye the next line can be something like "I went to the bank to cash my cheque" from *I Don't Give a F**k* by 2Pac). While this baseline has the advantage of always having coherent bars, it does not take into account that rhyme structure in rap lyrics are more complex beyond just matching rhyme schemes at the end.

5.4 Experimental details

We trained a many-to-one single-layered LSTM model with the following hyperparameters. Each epoch required about 10 seconds. Hence, we were able to pinpoint the optimal hyperparameters through hyperparameter tuning.

- Max epoch: 20
- Batch size: 128
- Sequence length (the length of sequence the attention should be applied to): 16
- Embedding dimension size: 128
- Hidden dimension size: 128
- # layers: 1
- Learning rate: 0.005
- Exponential learning rate decay: 0.98

As mentioned above, we reduced the data size used to train the Transformer models since our CUDA ran out of memory when the entire dataset was used. This was a result of testing the model and noticing the frequent occurrence of jargon and frequent inclusion of the word "lyrics", which was most likely due to low-quality data.

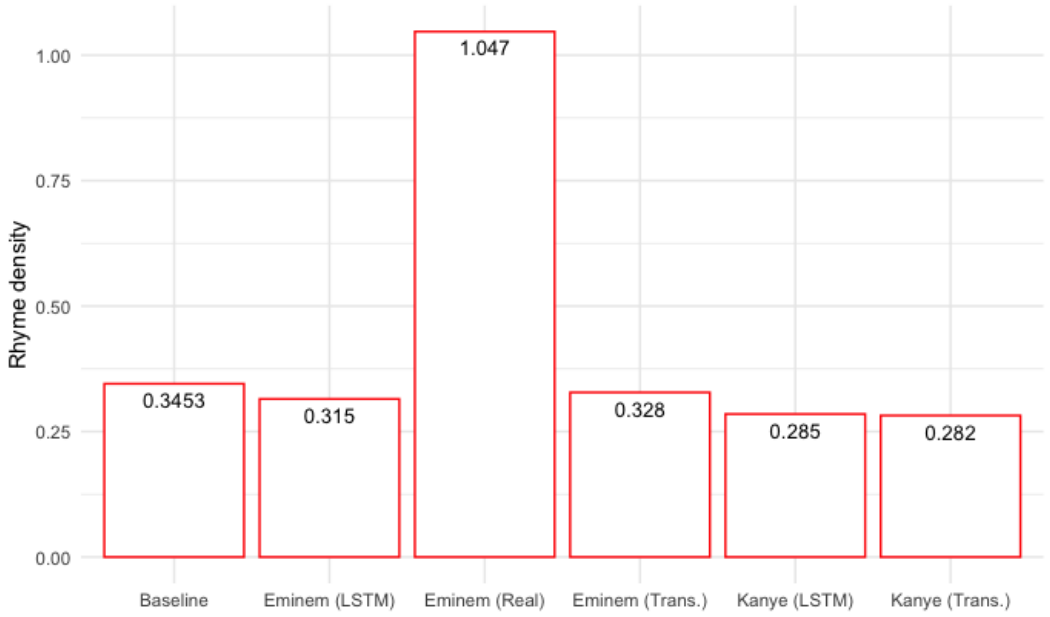
We also optimized for loss conversion within reasonable times (1 min per epoch) in our hyperparameter tuning, and that the following are the hyperparameters optimize for loss function:

- Max epoch: 15
- Batch size: 128
- Sequence length (the length of sequence the attention should be applied to): 8
- Embedding dimension size: 128
- Hidden dimension size: 512
- # layers: 1
- # heads: 4
- Dropout rate: 0.1
- Learning rate: 0.005
- Exponential learning rate decay: 0.98

We did not conduct formal experiments around hyperparameter tuning due to time constraints.

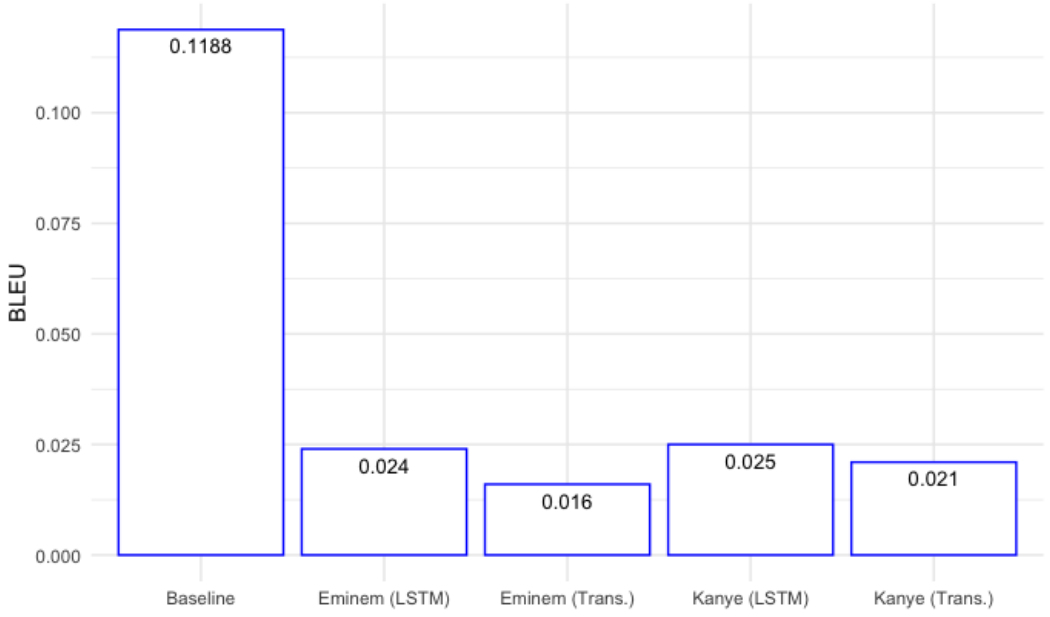
5.5 Results

The following are the average rhyme density scores for 100 lyrics generated by the baseline and our models as well as the lyrics Eminem actually wrote:



We found that all the models perform relatively similarly. Despite the baseline model almost "cheating" by retrieving lines that have rhyme directly with the preceding lines, it does not generate higher rhyme density as hip-hop can contain rhymes in virtually every part of the verse. We also find that none of the generative models (baseline included) are near the rhyme density of Eminem, indicating room for much improvement.

The following are the average BLEU scores excluding Eminem (since he would achieve 100 BLEU):



The results are consistent: the Transformer models does not outperform LSTM models based on quantitative metrics. The BLEU is much higher for the baseline, which can also be attributed to the fact that it "cheats" in its "bar" generation, and rap lyrics frequently share common words.

This is inconsistent with our original hypothesis that Transformer models will be much superior to LSTM. However, note that we trained LSTM with a larger dataset and for longer epochs. Also, due to

the aforementioned limitations of the evaluation metrics, qualitative evaluation of the generated lyrics is absolutely necessary. In the **Analysis** section, we will dive deeper into the qualitative analysis of the lyrics generated by our models.

Another interesting phenomenon that we noticed is that training for Eminem's lyrics for both LSTM and Transformer converged much more slowly than training for Kanye West's lyrics. We hypothesize that this is because Eminem composes very complex verses while Kanye keeps his lyrics rather catchy and simple (a noticeable trend, especially within his recent albums).

6 Analysis

For the sake of brevity, we will demonstrate the qualitative differences in the LSTM-generated lyrics and the Transformer-generated lyrics imitating Eminem. The following is an example of LSTM-generated lyrics:

LSTM Eminem
i can swallow a bottle of alcohol and i'll feel like godzilla it seems
a lane and he's always looked off my love holds
you can love you baby boy i can say i like it sent me
so i went to profit ah
i'll probably like a girl like a phone call
this game her business that's as i like it's just another i have is
x i got that adrenaline momentum
and for no sense of it it's an addict

The following is an example of Transformer-generated lyrics:

Transformer Eminem
i can swallow a bottle of alcohol and i'll feel like godzilla hit a valium
but there are back guess bein' weird's
i had to explain this shit i just put it to be a soldier
gonna go another song to me
where you and if you can see my dick and a blouse and the
you're flight so innovative was so won't keep winkin' and ran to the door
afford to the joe bring it lainie—uncle's i'm gonna can't be a stomach ammunition
and to the lawsuits soon as i'm lucky you

Both lyrics have a fair share of grammatically or contextually incoherent. However, the Transformer-generated lyrics have a line that can be salvaged such as "i had to explain this shit i just put it to be a soldier," which not only makes grammatical and contextual sense but also follows the Eminem's distinct lyrical style. We found this to be a consistent result – Transformer-generated lyrics almost always had a line or two that made grammatical and contextual sense, while LSTM-generated lyrics seldom had a line that made complete sense. Based on the fact that , we conclude that the Transformer models do a better job of generating lyrics that are coherent and follow the artist's unique style.

7 Conclusion

We have shown that Transformer based models are more effective than LSTM for hip-hop lyric generation even without any pre-training due to the use of attention scores. We have also found that the more complex a rapper's range of vocabulary and bar structure is, the harder it is to train a deep learning model around it.

As we described above, our rhyme density metric disregards unique rhymes which Eminem often utilizes. In addition, our quantitative metrics do not evaluate grammatical or contextual coherence of the sentences generated. We also have technical limitations in our models, especially for the Transformer model as we did not use a pretrained model. However, this was intended design to an extent to get accurate performance comparisons between LSTM and Transformer.

We aim to build more on this project by pursuing the following:

- Use pretrained Transformer model – this should lead to generation of rap lyrics that are much more coherent and the fine-tuning will allow it to follow the rhyme and flow structure of rap lyrics.
- Use larger dataset that is validated – it will lead to a larger vocabulary for the Transformer models.
- Have a larger group of human evaluators – because human evaluation is inherently biased, we hope to address it through having a larger group of evaluators.

References

[1] Eric Malmi, Pyry Takala, Hannu Toivonen, Tapani Raiko, and Aristides Gionis. DopeLearning: A computational approach to rap lyrics generation. In *Association for Computing Machinery (ACM)*, 2015.

[2] Jey Han Lau, Trevor Cohn, Timothy Baldwin, Julian Brooke, and Adam Hammond. Deep-speare: A joint neural model of poetic language, metere and rhyme. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, 2018.

[3] Language modeling with nn.transformer and torchtext. In *https://pytorch.org/tutorials/beginner/transformer_tutorial.html/*, 2022.

[4] Genius.com. In *https://genius.com/*, 2022.