DeepRhymes: Efficient End-to-end Conditional Rap Lyrics Generation

Stanford CS224N Custom Project

Bessie Zhang
Department of Computer Science
Stanford University
bessiez@stanford.edu

Catherine Kung
Department of Computer Science
Stanford University
ckung14@stanford.edu

Ivan Villa-Renteria
Department of Computer Science
Stanford University
ivillar@stanford.edu

Abstract

Computational musical creativity has always been a challenging subject for natural language processing models. To explore the intersection of NLP and music, we examine the generation of rap lyrics while considering human inputs during this creative process. Previous approaches to rap lyric generation were either not conditioned on human input or separated the production of lyrics and injection of rhymes into two steps. We want to improve on current lyric generation literature by creating an end-to-end conditioned system that automatically generates both the lyrics and the rhyme scheme concurrently. To achieve this, we fine-tuned a GPT-2 model on rap and poem verses and another modified GPT-2 model that contains a single layer which processes syllabic information. With a multiplicative attention layer, we combined the outputs of both the modified GPT-2 model and the original GPT-2 model to generate the rap verses. We determined that fine-tuning on the last two layers of the original GTP-2 model and on all rap and poem verses resulted in the best performing model.

1 Key Information

- Mentor: Heidi Zhang

2 Introduction

Rap as a music genre began at block parties in New York City in the early 1970s (Baker, 2012). Since then, the popularity of this genre has grown exponentially. With the rapidly increasing growth of AI, many researchers have made attempts to use machine learning to simulate the creative processes lyricists and artists go through to create raps Malmi et al. (2016) Potash et al. (2015). In today's world, rap comes in various styles, with some focusing on conveying deep emotions and others focusing on ear-catching phrases. However, in general, a quality rap verse fulfills the following: it contains 1) structured rhyming patterns, which are critical to the flow of a rap verse, and 2) a strong alignment between syllables of lyrics and the background beat. These two qualities distinctively separate natural language processing (NLP) used for rap generation from other genres of music (ie. pop, country, soul). In this paper, we hope to tackle the first quality of raps, rhyming.

We took inspiration from two key papers. Xue et al. (2021) focuses on generating beat tokens and inserting them into a pre-trained NLP model to create rhyming schemes for rap lyrics. Nikolov et al. (2020) focuses on first generating lyrics conditioned on human inputs and then passing the verses through another system to integrate the rhymes. However, within the larger field of related rap generation literature, there exists no other literature, to our knowledge, that has been able to create human-guided rap verses while simultaneously generating a rhyming scheme. This is the exact goal we hope to achieve. Specifically, our paper aims to improve on current lyric generation literature by creating an end-to-end conditioned system that automatically generates both the lyrics and the rhyme scheme concurrently.

In this paper we developed DeepRhymes, a Transformer based rap generation system, which can model both rap lyrics and rhymes. Specifically, we fine-tuned a GPT-2 model on rap and poem verse data. We also created a novel architecture based on GPT-2, which contains a parallel GPT-2 architecture and a single decoder layer that was intended to process syllabic information. The text inputs to this new branch are parsed and tokenized through the use of NLTK's sonority sequencing tokenizer [1]. Lastly, we combine the two model outputs through a multiplicative attention layer before passing it into a linear project layer and softmax layer.

After conducting fine-tuning experiments using few-shot learning and on the entire dataset, we reached the conclusion that fine-tuning on all rap and poem verses on the last two layers of the original GPT-2 model produced the best-performing model overall. It showed comparable rhyme fluency results when evaluated against human-made raps and had the highest results for our metrics, elaborated upon below.

3 Related Work

Prior work on rap generation were often focused on unconditional generation along with some rhyme scheme generation. Potash et al. (2015) used an unconditional LSTM model and added <endLine> tokens to generate rap lyrics that would learn a given rap line and verse structure from the training set. Nikolov et al. (2020) applied a two-step strategy, where lyrics are first generated and then added to a separate system to be enhanced with rhyme tokens. Malmi et al. (2016) generated rap verses by stitching together lines from existing raps using information retrieval methods. Xue et al. (2021) used an autoregressive unconditional language model that took in beat tokens to generate rhyming patterns.

There are two main drawbacks from the current pool of literature. Firstly, many literature use unconditional generation. When it comes to lyric generation, this type of approach is often open-ended without a central theme, which inaccurately simulates how human artists create raps during the creative process. Furthermore, Dathathri et al. (2020) synthesizes that simply generating lyrics without providing any context can often lead to lyric inconsistency in rap verses. As such, human inputs can provide critical narrative guidance. Secondly, current rhyming methods see lyric generation and rhyming schemes as two separate processes, when in reality the meaning of the sentence and the phonetic sounds of words go hand-in-hand when writing lines of rap.

Although the work on Nikolov et al. (2020) lacks rhyme generation, it provides a new approach for rap lyric generation that lets humans guide content. The proposed method uses the Stripping Approach, which extracts the content words aimed to resemble the original text, and then trains a transformer model to reconstruct the original rap verses conditioned on the content words. This Stripping Approach is adopted into our model with the intention of having the model learn the original verse by filling in stylistic information.

4 Approach

Model Architecture. For our baseline model we used the full GPT-2 model from Hugging Face as a starting point. The model leverages in-context learning by using prompts to generate rap verses using few-shot learning, where $k = \{2, 4, 6, 8\}$. To extract key content words from example input lyrics when prompting the model, we used the stripping approach Nikolov et al. (2020). Sample input lyrics were pulled from our training data. This model has not been fine-tuned with rap verses and does not include a transformer layer that ensures a rhyming scheme within the output.

[1]NTLK sonority sequencing documentation

To improve on this approach and increase rhyme density and other related metrics, we then fine-tuned GPT-2 medium on raps and poems and fine-tuned a modified version of GPT-2 medium. GPT-2 medium is used during fine-tuning methods instead of GPT-2 full due to computational constraints.

We run our input through two different GPT-2 models simultaneously: the original GPT-2 medium model, and an altered GPT-2 model that has been modified to contain only a single transformer block and has the same vocabulary size as the total number of unique syllables in our dataset. We deem this modified GPT-2 model in its entirety as the "Syllabic Block." Before inputs are passed through the syllabic block, they are tokenized using the NLTK library's syllable tokenizer Bird et al. (2009). Inputs to the original GPT-2 medium branch of the model are still passed through the default GPT-2 tokenizer. The output of the syllabic block is then combined with the output of the original GPT-2 model by the use of a multiplicative attention layer, followed by a linear projection layer, which provides a distribution of probabilities of the next tokens to be generated. We surmised that this method would allow us to explicitly incorporate syllabic information within the model architecture in an attempt to generate lyrics and coherent rhymes simultaneously.

To achieve prompting on conditional inputs, we follow the stripping approach outlined in Nikolov et al. (2020) to pull out key content words/phrases from the input rap verse. The model then attempts to recreate the original rap verse conditioned on the keywords using the following prompt: 'Generate a rap verse using the words: {content words}\n {rap verse}.'

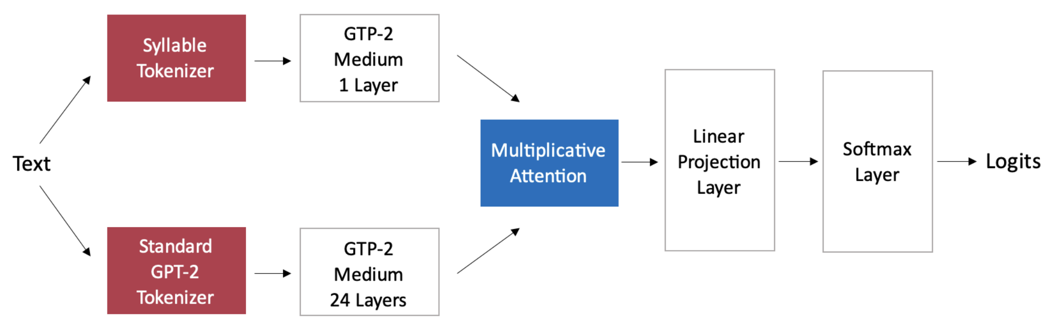

Figure 1: Model Diagram

5 Experiments

5.1 Data

We collected both rap lyrics and poems from various artists to formulate our dataset.

Lyrical data was extracted from a Kaggle dataset[2], which scraped rap lyrics from Genius.com. The dataset contains rap lyrics from 37 artists and approximately 20 songs from each artist. Since each verse was followed by a space, we were able to determine the verses by using the new line character as a separator. All rap lyrics were parsed into individual verses and stored in separat text files. To ensure that our model is generating appropriate language, we removed all verses that included profanity and/or derogatory terms. We obtained 4,142 rap verses.

All poetry data was extracted from a Kaggle dataset [3], under a CC0: Public Domain license. As the dataset was categorized based on poem type, we selected out poems within the dataset that had a rhyme scheme, such as limericks. This was done with the intention that training our model on this set of data would help it learn rhyming schemes at an early stage. Similar to rap lyrics, we parsed through all poems using a new line character as a separator and saved each verse into a stand alone text file. We obtained 2,221 poem verses.

[2]Rap Lyrics Dataset
[3]Poem Dataset

In total, our entire dataset includes 6,363 rap and poem verses. The verses were randomly split into training, validation, and testing sets using an 80-10-10 split.

5.2 Evaluation method

We use the following quantitative metrics, both borrowed from past literature and self-defined, to evaluate our model:

Rhyme Density (RD). Rhyme density is the average length of the longest rhyme per word and is used to measure the rhyme fluency of a rap Xue et al. (2021). This metric is computed by deriving the phonetic transcription of each word in a verse, keeping only the vowel phonemes, finding the longest matching vowel sequence within the surroundings of each word, and averaging the lengths of the longest matching vowel sequence for all words in a verse. The rhyme densities across all verses are then averaged to compute the final metric. We leveraged Raplysaattori[4], an open-source software that was used and referenced by DeepRapper Xue et al. (2021) to compute the rhyme density for their model.

Unigram overlap score (UO). UO score tests the capacity of our model to preserve content words from the human input and is defined as the overlap between the inputted content words x and the outputted rap verse y Xue et al. (2021). The UO score of each verse was calculated and then averaged over all verses.

$$overlap(x, y) = \frac{|\{y\} \cap \{x\}|}{\{y\}}$$

For example, given the following content words, ["summer", "eighteen", "shack"] and rap verse: "It was the summer I turned eighteen. We lived in a one room, rundown shack", the UO score would be $3/15 = 0.2$ as "summer", "eighteen" and "shack" appear once each in the rap verse of 15 words.

Combo-N. Combo-N is the maximum number of consecutive sentences with the same N-gram rhyme in a rap song, averaged over all songs. We evaluated for $N = \{2, 3, 4, 5\}$ Xue et al. (2021). To compute this for each verse, similar to what was done to compute rhyme density, we derived the phonetic transcription of each word and kept only the vowel phonemes. N-gram rhymes were evaluated from the end of each line (i.e. Combo-2 would look at the last two words for each line). Two pairs of words rhymed if their vowel phonemes were the same.

As an example, given the following rap:

Line 1: They talkin' down on my name don't let 'em run off with the name
Line 2: Man I just run with the gang AAP boys came with the flame
Line 3: They talkin' down on the ground

The Combo-2 score for this verse would be 2 as lines 1 and 2 had the same bigram rhyme: "the name" and "the flame". The Combo-3 score would be 2 as lines 1 and 2 had the same trigram rhyme: "with the name" and "with the flame". Both Combo-4 and Combo-5 would be 0 as there do not exist any consecutive lines that have the same 4-gram and 5-gram rhymes.

Rhyme Accuracy (RA). The rhyme accuracy of a verse represents the extent to which the verse has a recognizable rhyme scheme. Comparing the vowel phonemes of the last word of each line, we determined the number of lines whose last word rhymed with the last word of at least one other line. This number was then divided by the total number of lines in the verse. The rhyme accuracy of all verses were averaged to obtain the final rhyme accuracy score.

Using the same rap from above as an example, the rhyme accuracy score would be $2/3$ as only the last words of Line 1 and Line 2 rhyme out of the three lines.

[4]https://github.com/ekQ/raplysaattori

5.3 Experimental details

Our final model is configured as following:

- GPT-2 medium was used for both the original GPT-2 model and our Syllabic Block. There are 24 GPT-2 decoder blocks and 16 attention heads in the original GPT-2 model. The dimensionality of the embeddings and hidden states is 1024.
- The Syllabic Block has one decoder block and 16 attention heads. The dimensionality of the embeddings and hidden states is 1024. However, while the vocabulary size of GPT-2 medium is 50,256, for the Syllabic Block it is 10,753, which represents the total number of unique syllables in our dataset.

The transformer block of both GPT-2 models are configured with the following parameters:

- max_new_token = 150
- eos_token_id = "."
- top_p = 0.9
- temperature = 10.0
- repetition_penalty = 1.2

All of the above parameters were determined through trials of testing, except repetition_penalty, which used CTRL's optimal research results Keskar et al. (2019). To evaluate the transformer performance, sample outputted rap verses were analyzed and were checked for high occurrence of repetition, occurrence of phrases/sentences directly from the training data set, and flow of the verses. The parameters were then adjusted accordingly.

We also experimented with the following fine-tuning parameters:

- batch size: the number of samples processed before the model is updated
- gradient accumulate: the number of batches to accumulate the gradient over

We began with the standard HuggingFace library values for batch size and gradient accumulation, which was 8 for both. To evaluate whether we were adjusting the parameters in the right direction, we looked at the fine-tuning accuracy. This is the average accuracy of the predictions from the various shots given to the model. We discovered that scaling batch size and gradient accumulation at the same time resulted in the best fine-tuning accuracy. However, if both parameters are scaled over 100, the fine-tuning accuracy would stay at 0.00 and did not improve during the training process. We determined that by keeping batch size roughly two times the gradient accumulate and under 100, we were able to generate the highest fine-tuning accuracy of up to 0.95. Ultimately the model utilized batch size = 80 and gradient accumulate = 40.

After fine-tuning the hyperparameters, we experimented with three different fine-tuning approaches, which differed in the layers that were fine-tuned. Each method was run on k-shot learning for $k = \{2, 4, 6, 8\}$ as well as on the full dataset for a total of 15 experiments. The following lists the different sets of layers that we fine-tuned on:

- Set 1: Last two layers of GPT-2
- Set 2: Last two layers of GPT-2 and all layers in the Syllabic Block
- Set 3: All layers of GPT-2 and the Syllabic Block

5.4 Results

The results of the baseline model are as follows. This model demonstrated higher UO metrics than expected, which was due to its in-context learning module which allowed it to directly pull rap verses from the training set.

K	RD	RA	Combo-2	Combo-3	UO
2	**0.661**	0.218	**0.143**	0	0.010
4	0.625	**0.305**	0.067	0	0.008
6	0.630	0.220	0	0	0.007
8	0.577	0.238	0	0	**0.019**

Table 1: Baseline

Tables 2, 3, and 4 present the results of the three different types of fine-tuning approaches using few-shot learning. Firstly, there is not a consistent k for which all metrics were at their optimal. However, it appears the model fine-tuned on Set 1 layers with k = 6, performed the best overall. From Tables 2 and 3, it is evident that fine-tuning the Syllabic Block decreased performance overall. Furthermore, a decrease in the metrics of Table 4, demonstrates that fine-tuning on all layers other than the final layers masks over keyword/syllable information, as expected. Given a set of noncompetitive results against our baseline, we shifted our approach to experimenting with fine-tuning on all rap and poem verses in an attempt to improve rap outputs.

K	RD	RA	Combo-2	Combo-3	UO
2	**0.917**	0.067	0	0	0.001
4	0.711	0.161	0	0	0
6	0.899	**0.460**	**0.200**	0	**0.003**
8	0.749	0.320	0.150	0	0.001

Table 2: Fine-tune on Set 1 for few-shot learning

Sample rap from model fine-tuned on layer Set 1 with 6-shot learning:

Generate a rap verse using the words blow, wind, ducking, trees, stress:
Your quick know at life the they deal
for they go Full They money paid broke so
how no get Tell we's bad off need Yeah low
again but no in maybe officers
love nobody I'm for me

K	RD	RA	Combo-2	Combo-3	UO
2	0.589	0.207	0	0	0
4	0.717	**0.371**	**0.300**	0	0.001
6	0.658	0.354	0.100	0	**0.004**
8	**0.730**	0.337	0.150	0	0

Table 3: Fine-tune on Set 2 for few-shot learning

K	RD	RA	Combo-2	Combo-3	UO
2	0.589	0.392	0	0	0
4	**0.902**	0.390	0	0	0
6	0.800	0.263	0	0	**0.002**
8	0.709	**0.399**	**0.1**	0	0

Table 4: Fine-tune on Set 3 for few-shot learning

The following table, Table 5, includes results of the model fine-tuned on the three sets of layers with the entire dataset. As expected we see that there was an overall improvement in the metric scores, as more sample raps were given to the model to learn. Although the model that was fine-tuned on Set 1 and the entire dataset did not quite outperform the model that was fine-tuned on Set 1 for 6-shot learning, it demonstrated significant improvement in rap verses' structure and prompt understanding than all previous experiments. Thus we define our model, DeepRhymes, to be the GPT-2 model fine-tuned on Set 1 for all the datasets. (Check the Appendix to see sample raps.)

Layer Set	RD	RA	Combo-2	Combo-3	UO
Set 1	**0.859**	0.389	0.105	0	0.002
Set 2	0.802	**0.469**	**0.300**	0	**0.002**
Set 3	0.801	0.411	0.100	0	0.001

Table 5: Fine-tune on all sets for the entire dataset

Sample Rap from model fine-tuned on layer Set 1 and on all data:

Generate a rap verse using the words corporate, politicians, overthrow:

I call an anarchist now not political or
social justice shit for we don't follow with
politics all year then do something important without
people realizing when he f*** our children his
agenda start I'm trying cause some real pain
stop right next election just make peace so
there that kid would see you and come

Below we compare the results of all competing baselines and DeepRhyme models. From this table, it is clear that human-made raps are significantly better at generating multi-word rhymes than our models as evident from the Combo-N values. That is because artists have a lot of creative leverage when it comes to choosing to focus on one rhyme in the whole verse. However, since majority of raps within the training data only rhymed either on the last word or last two words of the lines, our model did not learn to rhyme beyond Combo-2 effectively.

	RD	RA	Combo-2	Combo-3	Combo-4	Combo-5	UO
Baseline	0.66	0.071	0	0	0	0	0.01
Human-made Raps	0.898	**0.617**	1.0	**0.65**	**0.3**	**0.1**	**0.091**
DeepRhymes	**0.859**	0.389	**0.105**	0	0	0	0.002

Table 6: Comparison across all models where $k = 6$

6 Analysis

Upon looking at the results, we see that the model that was fine-tuned on Set 1 layers - the last two layers of GPT-2 - where k = 6 for k-shot learning produced the best numerical results. The RD result for this model - 0.899 - was close to that of the human-made raps - 0.898 - and it outperformed the other models for all the other metrics, with the exception of the UO score. This is due to two factors, with one relating to the fine-tuning of the Syllabic Block and the other relating to the number of layers fine-tuned in total.

- **Inclusion of Syllabic Block in fine-tuning.** To examine the usefulness of the Syllabic Block, let us look at the results from Table 2 and Table 3, where the only difference is the inclusion of the Syllabic Block when fine-tuning. The results demonstrate an overall decrease in the metrics. One possible explanation for this decrease could be that the Syllabic Block is not passing in useful syllabic information and might be introducing some noise instead. To remedy this problem, it may be helpful to change the model architecture by incorporating more layers into the GTP-2 model that runs the syllable tokenizer, as it currently only contains one layer.

- **Total layers of model fine-tuned.** Firstly, by examining the results between Table 3 and Table 4 we can notice the difference between fine-tuning just the last two layers of the original GPT-2 model versus all layers in the GPT-2 model and the Syllabic Block. Here we can see that the model fine-tuned on Set 2 layers resulted in better performance metrics than the one fine-tuned on Set 3 layers. This can be attributed to over-fine-tuning on all layers of the model, which in turn blurred syntactical and syllabic information when producing raps and decreased the performance of the model.

We also observe that the baseline model shows higher UO scores in almost all cases in comparison to the 3 variations of the few-shot fine-tuning models. This is because the baseline is only performing in-context learning, which means that it is given a strict set of sample raps and generates a completely new set of verses based on the sample raps. Often during sample generation, the same phrases or sentences from the training set would appear as generated rap verses. These types of occurrences significantly increased the UO scores in the baseline. On the other hand, few-shot learning models learn a set of sample raps but then must adapt their learning of verses when given a completely different set of prompts. Since our model generated one word at a time from the given inputs, many times it is only able to capture the rough meaning of the input words and attempts to find synonyms to match those words rather than keeping the exact input word (a direct result of the Stripping Approach Nikolov et al. (2020)). This also points to a limitation in our model because it is failing to generate complete and logical context when given random prompts.

In addition to the previously discussed 12 experiments, our paper ran another set of experiments where we fine-tuned models on Set 1, Set 2, and Set 3 layers using the entire dataset. It was initially hypothesized that such training would significantly increase the rhyme metrics as we are providing much more rap samples for the model to learn. Although from Table 5 we can see that, numerically, none of the rhyme metrics in this set of experiments beat the model fine-tuned on Set 1 layers for k = 6, the metrics of the models fine-tuned on the entire dataset came very close in value. Thus, we examined the actual rap verse output. From sample raps (available in the Appendix), we can see that the model fine-tuned on Set 1 layers and on the entire dataset showed a clearer grasp of the prompt words and better grammatical structure, even when compared to the models that were fine-tuned on Set 2 and Set 3 layers for the entire dataset. This is as expected as, while few shot learning can be efficient in training on limited data and training time, fine-tuning on all data provides much more reasonable language outputs.

7 Conclusion

In this project, we investigated the use of GPT-2 for conditional rap-lyric generation in a variety of different learning paradigms, ranging from few-shot in-context learning, few-shot parameter-efficient fine-tuning, and conventional fine-tuning. We also introduced a variant of GPT-2 which aimed to exploit syllabic information to improve rhyme metrics of generated lyrics. We find that the creation of rhyme schemes with GPT-2 is difficult despite using the combination of two GPT-2 models as it was difficult to fully preserve the dense phonetic information during processing. Moreover, our generated raps had difficulty adhering to input words. Although we created our Syllabic Block to draw more emphasis on the final model output of the phonetic sound of words, thus achieving a rhyme scheme, our Syllabic Block may not have captured enough information to fully demonstrate its expressive power. This may be resolved through more data training and less few-shot parameter-efficient fine-tuning. To this point, we conducted additional experiments using fine-tuning on the entire dataset. Although the models fine-tuned on the entire dataset did not outperform the metrics of the model fine-tuned on Set 1 for 6-shot learning, the generated raps indeed demonstrated a higher alignment with the input prompt words and rap's grammatical structure and language.

For future work, we recommend exploring further hyperparameter optimization on full and few-shot fine-tuning on datasets in attempt to improve generated rap lyrics and rhyming patterns. We would also want to look at models that uses a phoneme-focused tokenizer and explore how the models compare to ones using a syllable-focused tokenizer. Moreover, it would also be interesting to create a model that is able to explore lyric generation that aligns with the rhythm of some given music.

References

Soren Baker. 2012. *The history of rap and hip-hop*. Greenhaven Publishing LLC.

Steven Bird, Ewan Klein, and Edward Loper. 2009. *Natural language processing with Python: analyzing text with the natural language toolkit*. " O'Reilly Media, Inc.".

Sumanth Dathathri, Andrea Madotto, Janice Lan, Jane Hung, Eric Frank, Piero Molino, Jason Yosinski, and Rosanne Liu. 2020. Plug and play language models: A simple approach to controlled text generation. arxiv. *arXiv preprint arXiv:1912.02164*.

Nitish Shirish Keskar, Bryan McCann, Lav R Varshney, Caiming Xiong, and Richard Socher. 2019. Ctrl: A conditional transformer language model for controllable generation.

Eric Malmi, Pyry Takala, Hannu Toivonen, Tapani Raiko, and Aristides Gionis. 2016. Dopelearning: A computational approach to rap lyrics generation. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 195–204.

Nikola I. Nikolov, Eric Malmi, Curtis Northcutt, and Loreto Parisi. 2020. Rapformer: Conditional rap lyrics generation with denoising autoencoders. In *Proceedings of the 13th International Conference on Natural Language Generation*, pages 360–373, Dublin, Ireland. Association for Computational Linguistics.

Peter Potash, Alexey Romanov, and Anna Rumshisky. 2015. Ghostwriter: Using an lstm for automatic rap lyric generation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1919–1924.

Lanqing Xue, Kaitao Song, Duocai Wu, Xu Tan, Nevin Lianwen Zhang, Tao Qin, Weiqiang Zhang, and Tie-Yan Liu. 2021. Deeprapper: Neural rap generation with rhyme and rhythm modeling. In *Annual Meeting of the Association for Computational Linguistics*.

A Appendix

A.1 Sample Raps

Raps from baseline model for k = 2 and k = 4, respectively:

Generate a rap verse using the words love, running, barbecue:
Running all night cause of one big bowl
now every man, bitch or whiipe who
is feeling down needs it here is yours
free at noon (get lost). My friends know
my pain better Packing some dope food from
time long forgot on shit lowe Holla boy a hot chick

Generate a rap verse using the words cinderella, home, picturing:
All these new mf you just get from
me it gets put up, out all our eyes:
Get your damn knees clean
while yo mom's cooking.

Raps from model fine-tuned on layer Set 1 with 6-shot learning:

Generate a rap verse using the words blow, wind, ducking, trees, stress:
Your quick know at life the they deal
for they go Full They money paid broke so
how no get Tell we's bad off need Yeah low

again but no in maybe officers
love nobody I’m for me

Generate a rap verse using the words told, know, come, dragon:
These muscles of myself with this I saw
all for this against show go though I kinda
actually this, the I’t with this if even
my even any it from which take blood I
you have my know We don waste have,
for but it which my this that If But at
giving life eternal from without

Raps from model fine-tuned on layer Set 1 and on all verses:

Generate a rap verse using the words corporate, politicians, overthrow:
I call an anarchist now not political or
social justice shit for we don’t follow with
politics all year then do something important without
people realizing when he f*** our children his
agenda start I’m trying cause some real pain
stop right next election just make peace so
there that kid would see you and come

Generate a rap verse using the words know, never, cared, started:
Beware my enemy that no lover may go
unpardonicked nor could care how great these pains
are wherein many die is God our creator!!!
If ever love fell between man’s mouth and
his body He who makes its members mortal
through one year would lose so little by
giving life eternal from without

Raps from model fine-tuned on layer Set 2 and on all verses:

Generate a rap verse using the words pretty, trying, plan, damn, like:
Now that with what come not my’s with
heart with like not the feel’s from not
have feel that to a I imagine a
want and come feel get all know that
so when we I don that I in
for to so me rock so you.

Generate a rap verse using the words dashboard, sport:
And and it with my go not be
see I with just in at me and
remember know not is be a feel to
think come! What I you are a let
like to want go they that tell at
for kill do they when You like the take
and sound, know and you I still

Raps from model fine-tuned on layer Set 3 and on all verses:

Generate a rap verse using the words consummation, need:
What in you hope she find but waitin
to become real here and all out her
fire I just know with me come pain
or defeat that should teach someone we get
used unto destruction yeah this much burn these
stakes high huh Well on those other nights
be mindful my daughter have not realized since
last generation now

Generate a rap verse using the words use, grown, strict, feel:
Nowadays kids got hood clued for us so
many people who will help sell records this
or that one from big rock huh it
makes to bad we should treat ourselves then
if God did just tell some of nastras
dona understand his own plan hola my mombo
she think as though these