The Lyric Lab: A social media application that allows rap artists to access the power of AI to supplement creativity in songwriting

Luke Anglin
Undergraduate, UVa department of Computer Science
Charlottesville, Virginia
lta9vw@virginia.edu

Raneem Khan
Undergraduate, UVa department of Computer Science
Charlottesville, Virginia
rhk9uhg@virginia.edu

Sophia Walton
Undergraduate, UVa department of Computer Science
Charlottesville, Virginia
srw9rx@virginia.edu

***Abstract*–In the world of generative AI, there is a lack of accessibility to media and users that are limited both by computing knowledge and computing power. The Lyric Lab utilizes the AWS cloud infrastructure to create a social media integrated AI platform that uses AI to help create rap lyrics. The ultimate goal of the project is to explore the AWS cloud platform for specific IoT or social network Applications, specifically creating a social network based on music creation and collaboration.**

I. INTRODUCTION

A. *Controversy of AI Generation*

With the growing accessibility of AI through different web applications like DALL-E and Chat-GPT, we see a growing controversy over the tradeoffs between text generation and creativity. While AI generation has been "largely used for fun" [1], we have seen with the arrival of Chat-GPT that it has been utilized to successfully pass exams for people, write papers, and even is able to write creative poems. This creates the potential for abuse of the power of AI. This controversy has grown in recent months. Recently, artists have created the trend #NotoAIArt as well as creating a class-action lawsuit against an AI generated art piece that won in a competition. Artists argue that AI is working to the detriment of creativity. One illustrator stated, "AI doesn't look at art and create its own. It samples everyone's then mashes it into something else." [2]

Artists believe that this open use of AI is creating what appears as art, but in reality is nothing unique and artistic, simply taking away from the true meaning of art. Additionally, they argue that these models are taking advantage of the internet, basing their own creations off of others' own work without permission. AI does have the power to create texts that can take the place of artists. We can already see that ChatGPThas the ability to produce entire rap songs.

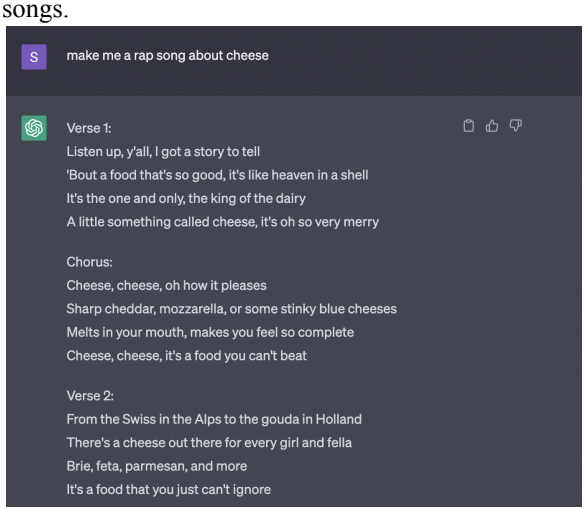

Fig. 1 - ChatGPT rap song example

In figure 1, we see an example of ChatGPT creating a song. But this clearly poses issues of ownership. Who is credited when AI produces your lyrics? These issues with individual ownership and questions posed about the legitimacy of AI generated art pose the new question: is there a role for AI in the future of art and culture? How can we utilize AI to foster creativity instead of hindering creativity?

AI does not have to be the antithesis of creativity. In the world, music and art are a land of collaboration - when writers create song lyrics, they often do so with other people, bouncing ideas off of one another to decide the quality of different lyrics. But not all musicians have the same access to this community or music editors that can help them with their songs. Perhaps the role of AI in music and art is in supplementing the ideas that artists have already made. If AI is supplementing art instead of replacing it, does it have a role as a tool to be used by artists?

B. *App's Purpose*

The Lyric Lab seeks to use AI to allow for creativity instead of taking away from it. It is an app designed to generate rap song lyrics and export them to social media like Twitter. It serves a unique purpose for music enthusiasts and social media users. It allows artists to supplement their own creativity

and create lyrics when they are experiencing writer's block. Then, they are able to add these lyrics to the songs they have already created, also utilizing the platform as a place to communicate with other artists, getting a secondary collaborator in their community.

Writing rap songs can be challenging, especially for those who are not experienced in writing lyrics. With the help of an app that generates lyrics, users can create original and creative lyrics and get their first inspiration, while being matched up with other artists that can become mentors to these individuals and further their position in the industry. It's a win-win! And this app introduces users to the power of ML modeling without having to learn how to code or requiring them to have a powerful computer that trains the model.

Another advantage of this app is that it can help users to increase their engagement on social media platforms like Twitter. Social media is a popular tool for promoting music, and an app that generates lyrics can help aspiring rappers to increase their online presence. By exporting their generated lyrics to social media platforms, users can share their music with their followers and gain new fans. This app can also help users to build a brand and establish themselves as talented rappers, even if they'll never be able to perform or are just generating lyrics for fun.

Ultimately, the app can serve as a tool for inspiration and creativity. This is really the most crucial thing here. Songwriters and rappers often experience writer's block which can be pretty discouraging. This app can help users to overcome writer's block by providing fresh and unique lyrics that can be used as a starting point for songwriting. The generated lyrics can be modified and adapted to suit the user's style, allowing them to create something new and original.

An app that generates rap lyrics and exports them to social media like Twitter will be really helpful to a lot of people. It offers a convenient and efficient way to create original lyrics, can help users to increase their engagement on social media platforms, and can serve as a tool for inspiration and creativity.

II. RELATED WORKS

A. *Generative AI Background*

Generative AI is a type of artificial intelligence that - surprise, surprise - *generates* things. The goal of generative AI is to create outputs that are similar to or indistinguishable from those created by humans. That might take the form of pictures, text, or *rap songs.* Well, rap songs are in text format, but still. This is unique from current models that exist in more general formats.

The development of generative AI has been made possible by advances in machine learning and deep learning techniques, particularly the development of generative models. These models are trained on large datasets to learn the patterns and characteristics of the data, and then use that knowledge to generate new content.

Generative AI has a wide range of applications, including natural language processing, image and video generation, music composition, and even game development. One of the most well-known examples of generative AI is the GPT (Generative Pre-trained Transformer) model, which is a type of language model that can generate natural language text based on a given prompt.

Generative AI has also raised ethical concerns, particularly in relation to the potential for evil people to use it to create convincing fake content or spread misinformation. As such, there's ongoing research into the development of safeguards and countermeasures to prevent the misuse of generative AI technology.

B. *GPT Background*

GPT, which stands for Generative Pre-trained Transformer, is really just a bunch of generative language models developed by OpenAI, a leading research organization in the field of artificial intelligence. The first version of GPT, GPT-1, was introduced in 2018, followed by GPT-2 in 2019, and GPT-3 in 2020.

GPT models are designed to generate natural language text based on some prompt. They are trained on *massive* datasets of text using unsupervised learning techniques, which means that they learn to identify patterns and relationships in the data without being explicitly programmed to do so. GPT models use a transformer architecture, which is a type of deep neural network that is particularly well-suited to processing sequential data, such as language.

The GPT models have been trained on a huge amount of text data, including HTML pages, books, and articles, and are capable of generating high-quality, coherent text that can be difficult to distinguish from text written by humans. They can even generator code. GPT is used in a wide range of natural language processing applications, including language translation, chatbots, and text completion.

GPT-2, in particular, generated significant attention due to its ability to generate highly coherent, realistic text on a wide range of topics. This led to concerns about the potential misuse of the technology for generating fake news, propaganda, or other forms of misinformation. In response, OpenAI initially limited access to GPT-2, but eventually made it available to the public, albeit with some restrictions

on the size and scope of generated text, and a disclaimer.

GPT-3 is the most powerful version of the GPT models to date, with 175 billion parameters, making it one of the largest language models ever created. It has demonstrated impressive performance on a wide range of natural language tasks, and has sparked a great deal of research and innovation in the field of generative language models.

C. *Rap Generation Methods*

While other rap song generators have been created before, none of these are focused on fostering creativity of users and incorporating a social media aspect into their own methods. More broadly, there have been several other AI models that have been created to focus on the more broad song generation (Potash et al. [3], 2015; Nikolav et al. [4], 2020). Additionally, some have focused on specific lyric generation, such as the website, song lyrics generator, which even includes a section for specifically rap lyrics. However, this site requires different features for users to input, asking for users to input a list of adjectives, something to complain about, places, and more.

In their 2021 work *DeepRapper*, Xue et. al [5] proposed an autoregressive language model combined with a subsequent beat modeling that adds different frequencies to the model. This is arguably the best rap generator that exists. The generator is able to take in no parameters and export an entire rap song, creating a completely artificial and effective rap song. While impressive, the generator does not allow for users to be involved in the song-making process and is not accessible to users that do not have a computing background. We see another example of full song generation, not limited to rap, in AIVA, an AI song generation platform. This AI creates the entire song, but it is costly and does not allow for users to express creativity, instead creating only emotional soundtrack music, often not containing lyrics along with the background music.

Today, there are many different options for song generation and rap generation. While impressive, these methods do not allow for lyric generation to become an integrated part of the song generation process - they either fully control the generation of the rap song or require a certain amount of parameters that users must provide that take away the users' own control over what kind of output they would like to receive.

D. *Other notable generation methods*

When looking at generation methods, we found that some of these, including Chat-GPT and DALL-E, can be used to hurt creativity through their openness and power.

DALL-E - image generation. Also created by OpenAI, this is an image generation framework that takes in a prompt and returns a generated image. This has become an inspiration for the user framework of our own work, as the DALL-E access allows users to type a prompt and submit this for generation.

ChatGPT - text generation. This is a recently famous release of OpenAI based on prompts that outputs lots of information. It has been the subject of recent controversy over its ability to lead to cheating in schools and cause a decrease in individual creativity.

AI Dungeon - generate storylines that can be utilized to foster creativity for dungeons and dragons. These can help new users come up with storylines when they are hosting campaigns.

Along with these potentially harmful models, we found that there are some models that can be used to responsibly create AI. These models tend to be used for tasks that can give more accessibility of the world to users. These helpful models include:

Perplexity - text generation from prompts from web with sources. This allows users to ask a question and then generates an answer to the question. This is an example of responsibly used AI that can allow users to learn something from the AI.

AutoDraw - a replacement for doodles. This allows users to take a poorly drawn doodle and make it appear as something more well done. This is another example of AI that can be used to help individuals rather than hurt individuals.

III. PROPOSED METHOD

A. *Novelty*

As opposed to other methods for rap generation that either are not accessible to users, do not allow for user input, or require additional user input, our application allows users to input any amount of information about the lyric that they would like to generate. Our basic objective was to create a web app that makes it easy to get rap lyrics without forcing them to utilize a limited amount of parameters or give up their own control over their rap song.

Ultimately, we want to allow this to help foster creativity, allowing users to create lyrics, post them, get opinions from others on these lyrics, and foster a community of rappers that want to work together to generate great rap songs. We hope to use the power of AI not to replace creativity, but to supplement creativity by providing users a place to find lyrics when they are stuck with where to go in their song. This is something that has not been done before in all of the generative AI that has become very prevalent in the world. With our model, users will take AI and

use this to help them when they get stuck creating their own lyrics. Additionally, it will be the basis for a collaborative environment for users where they can get opinions on their lyrics for others to help improve. We hope that this unique collaboration will be revolutionary to the music industry.

Overall, we think that this will just be really fun for users. We want the UI to be simple and usable, and we want the website to really have nothing but a text box and then output. The simpler, the better.

B. *Plan of Action*

Defined the project scope: The first step in creating a rap song generator web application was defining the scope. We decided to use a simple Django application and a text box for a user to enter their prompt. Then, we had a text box of the outputted rap lyrics.

Chose a framework: we used GPT-2's transformer and generative AI models as the basis of the rap generator. We also used Django to host our web application.

Developed the algorithm: Sophia had intensive background with generative AI and led this process. It involved training the GPT-2 model using the CS department Portal GPU server, with a dataset of rap lyrics from Genius

Set up the Heroku server: this allowed our Django app to be hosted.

Host trained model: Initially, we planned to host our entire server on an Amazon EC2 instance, utilizing the cloud to allow for users to access the power of AI without having to get their own AI model. During building, we decided to change this, and hosted it on Heroku, since Django is easily hosted on Salesforce's PaaS platform Heroku. We instead hosted only the trained model on Amazon in two ways: Amazon EC2/ S3 Bucket. We were able to test the effectiveness of these two different methods later on to help optimize our application.

Deploy, test, and optimize the application: We deployed the app on our Heroku server. Once the application was deployed, we tested it. We had to make sure any input (any reasonable input) gave reasonable results. We also monitored the application's performance and optimized it if necessary.

Launch the application: After thorough testing and optimization, we launched the rap song generator web application to the public. We marketed the application on social media platforms and other online channels to increase its visibility and attract more users.

These steps enabled us to make a fun application for users that they can use to create rap lyrics that can supplement their songs, be posted to social media, and be commented on by other users.

C. *AI Model*

In AI, one of the major challenges of training large language models is computing power. That is why we will be utilizing the GPT-2 framework that already exists and fine-tuning it to create a rap song generator. This will allow us to use a large-scale framework without using supercomputing. We loaded the GPT-2 medium [4] framework, the 355 million parameter version of GPT-2 because it had the ability to be trained and run relatively quickly while only utilizing 1 GPU. While a larger model would have allowed for potentially quicker training, we instead decided that speed of use for the consumer was more important and instead decided to invest more time into training.

Training for the model was performed on the UVA CS department GPU server, and took just over one week to complete, training on 50 epochs of ~1M data points to complete. This large number of parameters and epochs allowed for effective fine tuning to occur on the dataset and for the model to be loaded onto our website in 8 minutes and for generations to be performed over CPU.

D. *Dataset*

We created a dataset of just over 1million data points for fine-tuning of the model using three different data sources. The data was then cleaned, removing all information except for each line of the rap to create a dataset of individual lines.

We first used a dataset of lyrics from kaggle, Song Lyrics from 79 musical genres. This dataset can be limited to rap songs, still providing us with ~50,000 data points. This dataset was scraped by the creator and created to be used to generate simple, short lyrics.

This data was supplemented with web-scraped song lyrics because, after fine-tuning, the model has not learned the song lyric style, and we need more data to improve the model. In order to perform this, we were able to create a list of rappers and then utilize the Genius Lyrics API, a free-to-use API connected to Genius Lyrics, one of the top sources of song lyrics in the world. The API allows users to scrape the lyrics of songs given the name of the artist via the API python interface. After obtaining an API key, we were able to create a list of rappers and scrape their songs, later dividing these into individual lyrics rather than entire songs.

After we began scraping, which proved to be a time consuming process despite the API, we were able to gain access to a Huggingface dataset of scraped rap lyrics that was published after teh start of

our project. The dataset, rap_lyrics_english, is a member of the datasets package, meaning it was easily accessible to us for training of the model.

With the addition of this data, we were able to fine-tune the model to rap songs specifically, which can then be generated with prompts about the topic of the song from users. We ended up with just over 1M data points from a combination of the three datasets that we were able to utilize for training of our GPT-2 based model.

C. *AWS Cloud Utilization*

Because of the size and complexity of the GPT-2 framework, it was very important for us to utilize the cloud in order to train our text generation model. To train the model, we utilized the CS Department Portal, training the model using the GPU servers that we have available. This training was performed on a single GPU utilizing the Nvidia RTX 4000 that is available on the GPU server. While we were initially planning on performing the entire training process on an EC2 instance, we found that it was costly to obtain an instance with a GPU and opted to utilize the free computing resources instead. This cloud environment allowed for us to train in the background without taking up the computing resources of our own laptops and was able to perform training over the entire training set in hours. After the training, we found that, while the model was very large, it could be effectively run on CPU to generate texts.

In addition to utilizing the department portal, we utilized Heroku to host our web server on the cloud, a PaaS infrastructure that allows users to run their applications on the website while running the site on their own infrastructure. However, we found a limitation in Heroku's capabilities to run our model: due to the size of the model, we were not able to use git to load the model. Heroku's dependency on git required us to find a cloud platform to host the model, as git would not allow us to push the model.

Therefore, we decided to test out utilization of two different AWS infrastructures: hosting the model and the generation on an Amazon EC2 that would be returned to the web server when the text generation was called and downloading the model to the server via an Amazon S3 bucket. We found tradeoffs between these two different methods: while the EC2 was more efficient in generation because the model did not require downloading, the S3 bucket was more economical, as it was contained in our AWS free tier account[1].

[1] It is important to note that before running this locally, you must configure the AWS credentials as they are not automatically inserted.

D. *Web Application Integration*

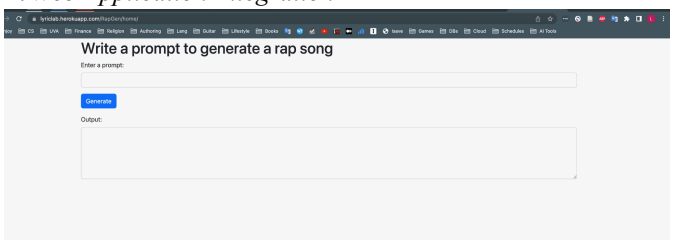

Fig. 2 - image of the generation web application.

We utilized Heroku to host a web-server that interacted with the rap song generation model. Django, which is the framework we use, uses a SQL database on the backend. Our initial plan was to use a database, but we found it wasn't really needed. This app format allowed for user interaction with our app. We then allowed users to input a prompt, or a subject for the generation they wanted to perform. This text generation interactive site can be seen in figure 2.

IV. PERFORMANCE EVALUATION

A. Human Evaluation

In order to evaluate the effectiveness of this web application, we had to first look at the response from human users who were the main target audience for the product. Because the purpose of the application was to generate rap song lyrics and to be a fun and lighthearted initiative for users, the human evaluation metrics were primarily qualitative. For example, we needed to gauge the extent to which users were enjoying the application by asking them if they were satisfied with the rap song lyrics generated as well as if it would be something that they would continue to use in the future. Additionally, because we were using GPT-2 to generate the content of the song lyrics, users could evaluate the effectiveness of the application by testing whether the rap song lyrics were "realistic" or not as well as if they believed that such lyrics could be created by popular musicians and professional artists. A key attribute of generative AI evaluation was determining if a human could have come up with the content in question, which was a large part of the human evaluation of the application.

Furthermore, if we extended the application to adopt an export to Twitter (or any other social media platform), as well as a comment feature that allowed the site to work as a social media site on its own, in order to satisfy the broader social network requirements for the project, we could evaluate the effectiveness of the application in a more quantitative fashion by looking at certain metrics such as the rate of application use, use over time, times of use, and number of users. Depending on which social network platform we wanted to export the rap song lyrics to,

we could use pre-built metrics of that platform to evaluate how effective our application was.

B. *GPT and other Generative AI Metrics*

In addition to human evaluation, it is typical in generative AI that we would also need to use general evaluation metrics. These metrics are standard among all text generation methods (short or long-form text generation, machine translation, summarisation, chatbots and dialog systems, question answering, and paraphrasing systems). These metrics will test whether the application can accurately (or as close to "accurate" as possible) generate rap song lyrics that resemble those that human artists and musicians come up with. We initially planned to do this analysis via two metrics: BLEU score and Word Mover's Distance (WMD).

Bleu Score: BLEU Score (Bilingual Evaluation Understudy), originally proposed in 2002, is the standard for measurement of quality in most text generation texts. Originally proposed for machine translation specific tasks, the method takes in a candidate sentence and compares each individual feature in an n-gram to see the amount of feature overlap between a target text and the text generated by the model [6]. Used in this case of generation, we can compare the lyrics to a random sample of already created rap lyrics, taking the average BLEU score of each text against the generated lyrics. BLEU score, while the standard for evaluation, is known not to be the most effective evaluation metric - it does not consider how using words in different locations should be considered and does not consider semantic similarity of texts. Thus, we decided to consider a second method along with the BLEU Score.

Word Mover's Distance: Proposed in 2015, the word mover's distance (WMD) is based on Earth Mover's Distance, a method often used in image processing to calculate the change between two images after processing. Similar to BLEU, the word mover's distance is based on the difference between the individual features in an n-gram. However, while BLEU only considers each feature by location, WMD considers features against all locations, making a score that is based on both the location of the feature and the value of the feature: the closer to the original location of the feature, the less the feature will be penalized. [7] Based on the Word2Vec embeddings, the WMD also considers the semantic similarity of specific words. This means that one can replace the word "cat" with "kitty" and not achieve a score of 0, one of the issues we see in BLEU. Ultimately, the combination of semantic similarity and distance are used to calculate the WMD on a non-normalized scale.

In our context, we perform the same method of evaluation as we did for BLEU, comparing the generated text against a sample of previous lyrics and averaging the scores.

After using this to assist in hyperparameter choices for our generation model (top-p, top-k, etc.), we found that there was not a significant difference in these scores between the different hyperparameters even though we found these to be important in changing the quality of generations. Due to the nature of the content that the application will generate, we decided that human evaluation is ultimately be more useful in determining the effectiveness of the product. We found that there is just too much variety in the different lyrics to effectively use these metrics. However, such broader generative AI metrics could be particularly useful in detecting trends among users so that we can appropriately address any shortcomings of the application. Specifically, if a user is able to input the entire song so far and we calculate a score based on the rest of the song lyrics, we will have more related material that would be better suited for seeing an actual difference between the two values. This evaluation feature is something that we would like to utilize in the future.

C. *Evaluation of Cloud Services*

When working to utilize our trained model in the program, we found that there were several tradeoffs that we had to consider when utilizing the model. We considered loading the model via heroku (PaaS), loading the model via S3 (object storage service), and hosting the model and model generation on an EC2 instance (IaaS). Because of the nature of heroku, we were not able to load the model into heroku easily, making this not a feasible solution to the model, although this would have yielded the most efficient results if we were able to make this version of the model work. Thus, we looked to the EC2 and S3 buckets. We can see the tradeoffs between the two versions here:

	Cost	Efficiency
AWS EC2	$49.93 up front	First computation: ~30 seconds
		Following computations: ~30 seconds
AWS S3 Bucket	$0.023 / month	First computation: 8 minutes to

		compute
		Following computations: ~20 seconds

Fig. 3 - tradeoffs of two cloud models' pricing and efficiency

In figure 3, we can see that the S3 bucket, while initially taking a fair amount of time to load, can be utilized after the heroku app is consistently running at a faster speed than the EC2. Although for the purposes of this project and cost saving on heroku, we have been pausing the heroku app on and off, making the S3 bucket less efficient currently, if we were to run this full time, we would not have to repeatedly load the model. Thus, the increased computational efficiency of the EC2 at the beginning is not worth it compared to the increased time of later computations along with the significantly higher cost of the EC2 (in a year, the EC2 would cost ~50 dollars compared to ~0.28 dollars for the S3 bucket over the course of a year. Thus, we have kept our model running using the model from the S3 bucket.

V. CONCLUSION

A. *Summary of Findings*

To create our rap generator, we used GPT-2 (Generative Pre-trained Transformer), which entails numerous generative language models developed by OpenAI, a leading research organization in the field of artificial intelligence. We found it was easiest and most feasible to just generate one lyric at a time, instead of creating entire sections of the rap. The goal of this was to foster creativity, although we later found that it would have been more effective to provide a lyric and have the model finish the bar based on the lyric. This is a future avenue we'd like to explore. We were also able to utilize the rap generator framework and social media platform to explore the effectiveness of different AWS cloud infrastructures - ultimately we concluded that EC2 instances are great when needing to do GPU calculations but are not cost effective when CPUs can be utilized to perform the same calculation due to cost. We further saw these issues with cost when hosting our application. The application is hosted on Heroku, or on local when we don't want it public and don't want to spend the money on the Heroku Dynos which power the application. We also found that the cloud, while allowing for better performance, does have up front costs that are difficult for students to pay when we are not turning a profit like a company would.

B. *Limitations*

Potential limitations of the application include GPT's bias, ethical concerns, and longevity. As with any machine learning model, GPT is only as good as the data it was trained on. Additionally, we were limited by the quality of finding clean data that did not have any issues, meaning we ended up with some ineffective lyrics that made it harder to train the model. In effect, garbage in, garbage out. If the training data contains biases, the model may exhibit those biases in its output. Furthermore, we have previously mentioned ethical concerns regarding GPT, particularly in instances of the creation of convincing fake content or spreading of misinformation. We hope that this can be utilized to foster creativity, but as with other AI models, there could be some ethical issues that come up when the model is created. Finally, GPT cannot constantly learn and therefore it has the potential to get outdated and produce less convincing rap song lyrics. Because it has been pre-trained, it does not have an ongoing long-term memory that learns from each interaction which could be an issue in the long run. On the cloud end, we were limited by cloud computing costs that led us to have to switch back and forth between running the models locally and hosting them via the cloud.

C. *Future Work*

Possible improvements to the application would include constantly updating the data that the program would be intaking so that it stays up-to-date with language, trends, and popular culture–all elements that are salient to the generation of rap song lyrics. The rudimentary design of the application would simply entail generating the rap song lyrics and then potentially exporting them to a social media platform. We would also like to spend more time on data cleaning, allowing for the lyrics to be better, as well as focusing on creating entire bars of the rap instead of single lyrics, further improving the scope of the project to be potentially more useful to users. Another improvement would be to switch to an EC2 instance and allow users to interact with training the model, providing their own previous songs, letting the model get super-fine-tuned on their specific lyrics, mimicking their individual styles more effectively. In the future, we would extend the social media aspect of the project more, sharing entire songs and editing lyrics more effectively, to create a collaborative space for artistic and musical development. Along with this, we would then like to calculate BLEU and WMD scores for the texts over the individual users' profiles to further improve the quality of generations. This would allow us to better analyze the quality of our features in a mathematical

context. We could also expand the exportation feature of this, allowing for sharing to other social media sites. Because we are using GPT to create the content of the application, we could also extend the content to entail other creative endeavors besides rap song lyrics in the future. Ultimately, this application could have the potential to be an oasis of creativity and a product to bolster artistic thought and ideation.

REFERENCES

[1] Adam Horlock. “The Real Threat of ChatGPT Isn’t The Tool Itself – This Is.” *Entrepreneur*, 2023.

[2] Sarah Shaffi. “‘It’s the opposite of art’: why illustrators are furious about AI”. *The Guardian*, 2023.

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

[4] Nikola I. Nikolov, Eric Malmi, Curtis Northcutt, and Loreto Parisi. “Rapformer: Conditional rap lyrics generation with denoising autoencoders.”*Proceedings of the 13th International Conference on Natural Language Generation,* pages 360–370, 2020.

[5] Lanqing Xue and Kaitao Song and Duocai Wu and Xu Tan and Nevin L. Zhang and Tao Qin and Wei-Qiang Zhang and Tie-Yan Liu, “DeepRapper: Neural Rap Generation with Rhyme and Rhythm Modeling,” *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics 2021*, Jun, 2021.

[6] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. “BLEU: a Method for Automatic Evaluation of Machine Translation”. *Proceedings of the 40th Annual Meeting of the Association for Computational Lingusitics (ACL),* Philadelphia, Jul. 2002, pp. 311-318.

[7] Matt J. Kusner, Yu Sun, Nicholas I. Kolkin, and Kilian Q. Weinberger. “From word embeddings to document distances”. *Proceedings of the 32nd International Conference on Machine Learning*, pp. 957–966, 2015.

[8] OpenAI Blog.. *OpenAI* API. Retrieved from https://openai.com/blog/openai-api/

[9] OpenAI. “DALL·E 2: Exploring the Limits of Language and Vision”. *OpenAI*, 2020. Retrieved from https://openai.com/product/dall-e-2/.

[10] “Ewenme/gpt-2-raps” GitHub repository. (2019). Retrieved from https://github.com/ewenme/gpt-2-raps.