Generating Lyrics Using Deep Learning Techniques

1st Shrivatson Ramaratnam Giridharan
Department of Applied Data Science
San José State University
San Jose, USA
shrivatsonramaratnam.giridharan@sjsu.edu

2nd Syama Ravi Teja Jerrypothula
Department of Applied Data Science
San José State University
San Jose, USA
syamaraviteja.jerrypothula@sjsu.edu

***Abstract*—Writing lyrics is a passion-filled kind of art. Since there are several variables to take into account while writing song lyrics, such as rhythm, rhyme, style, tone, and emotion. In this project, we investigate the use of deep learning to song writing. We take a dataset of lyrics from various melodic songs to train a recurrent neural network, and we then use the trained model to create new lyrics. We utilized models with LSTM and GRU, and these models are able to understand the linguistic structures and patterns used in song lyrics, using this information to produce new and creative material. To provide the models a better idea of how to preserve a certain writing style, we taught them at the character level rather than the word level. we discover that it can produce lyrics that are logical and consistent with the training data. We evaluated the proposed models with the metrics accuracy and perplexity metrics. Overall, our findings show how deep learning may be used to generate lyrics and aid composers in their creative processes.**

***Index Terms*—Deep Learning, Sequential Processing, LSTM, GRU, Lyrics Writing**

I. Motivation

A type of art that has enthralled people for millennia is making music. It takes a special and potent talent to be able to evoke feelings and deliver a message using a mix of tunes and words. Neural networks, which can be trained on a sizable dataset of lyrics to understand the patterns and structures of language, may be used to create music. There are several benefits to creating lyrics using neural networks. One benefit is that it enables the production of original and distinctive content. Neural networks are capable of generating novel word and phrase combinations that were maybe never considered before, resulting in the writing of original and creative lyrics. The ability of neural networks to learn from a variety of instances also allows them to produce lyrics in a variety of genres and styles.The ability for composers to save time utilizing neural networks for lyrics creation is another advantage. A neural network may instantly produce a variety of possibilities for a composer to pick from rather than forcing them to spend hours or even days attempting to come up with the ideal lyrics. This can give the lyricist more time to work on the melody or arrangement of the song.

In conclusion, songwriters may find it useful to generate lyrics using neural networks. Additionally to saving time and effort throughout the creative process, it may aid in the creation of fresh and distinctive content. Neural networks may assist in producing beautiful and powerful music by utilizing the power of machine learning.

II. Background

Music has been evolving with us for many decades. The sound of music has had a significant influence on society at large. It has altered everything, and it has also had a significant influence on the development of our culture. When we examine the origins of recorded music, we will observe that they were echoes of sounds found in the natural world. Both tones and repetition demonstrated this. Naturally, traditional instruments were also performed in a manner that was very comparable to the sounds of nature. The evolution of songs came when music and lyrics came together. At first the lyrics were also based on nature. Indigenous societies had a tight connection to the environment and to wildlife, which was frequently represented in the lyrics. Of course, this link was lost over time by contemporary culture. Now that the internet world has become more prevalent, we will soon notice that music is completely different. Currently there are various genres of songs such as rap, hiphop, melody, pop, country. Lyrics writing is one of the artful skills that only a few people turn it into a passion. According to the website of recruiter an average salary of lyricist falls in the range of $80,000 to $120,000. It is a creative work with appropriate meaning, style, rhythm which also should fit into particular category like Rap, Melody, Country. As of now, there is no substitute for human generated lyrics.

The use of neural networks for lyrics generation is a relatively new application of machine learning technology. It builds upon the longstanding practice of using computers to assist with the creative process, such as in music composition and other forms of art. In the case of lyrics generation, a neural network is trained on a dataset of lyrics from a variety of songs. This can include lyrics from different genres, styles, and time periods. The neural network then learns the patterns and structures of language used in the lyrics, and can generate new lyrics based on this knowledge. The use of neural networks for lyrics generation has gained popularity in recent years, as the technology has advanced and become more accessible. We strongly believe that deep learning can be one such powerful substitute for lyrics generation which is inexpensive and is scalable.

III. Literature Review

Eric Malmi et al. proposed a neural network model called Deepbeat which consists of MLP layers, concatenation layers.

In this paper the author used rap lyrics as they are with short sentences but they will be with a certain rhythm and rhyme. It is the reason author defined three parameters i.e., rhyme, structural, and semantic features. Rhyme density will be calculated by seeing the vowels relations in the words of a line. Semantic and structural are the parameters calculated by referencing the original human written lines. Author used RankSVM algorithm and combined these three parameters and evaluated the model developed. The developed model was able to predict the next line of the song successfully with the accuracy of 17% which is better than random selection [1].

Aaron Carl el al. said that rap lyrics broke the traditional structure and spelling of words in the English language and were more rhythm and rhyme oriented. Even for this, few spelling of words are modified in the lyrics for better pronunciation with rhyme. It is one of the reason the author used a character level generator rather than going with a word generator. In this paper, the author used recurrent neural networks like LSTM and GRU apart from Deepbeat network. GRU outperformed the LSTM and RNN interms of quality but it taking more time to train the model than the other two models [2].

Liu et al. (2019) provides an overview of the various approaches that have been used for lyrics generation using deep learning, including RNNs, transformers, and hybrid models. The authors discuss the strengths and weaknesses of each approach and the challenges and limitations of current methods. They also suggest directions for future research, such as incorporating additional linguistic features or leveraging external knowledge sources [3].

Fan et al. (2020) proposes a hybrid model that combines an RNN with a transformer for lyrics generation. The authors demonstrate that the model is able to generate lyrics that are more human-like and show a better understanding of the structure and content of songs compared to previous methods. They also show that the model is able to generate lyrics with a higher level of coherence and more consistent rhyme and meter compared to previous methods. The authors suggest that the hybrid model may be more effective at capturing the complex structure and content of songs than other approaches [4].

Many authors are using the rap lyrics for lyrics generation because of the small sequences with rhyming words and the words being repeated in the genre. But in this project we are considering only melody songs which are generally with longer sequences than rap songs. So, our trained proposed model will have the ability to generate lyrics with long sequences/ sentences. Apart from that most of the authors in the above literature survey, used MLP architectures and also normal LSTMs and GRUs. In this project we are proposing the model Bidirectional layer which learns contextual information from both past and future words in the input sequence since it examines the input sequence in two ways (forward and backward). It can be an advantage for next character generation task.

IV. Methodology

A. Data Collection

The lyrics are available from a wide variety of sources on many platforms. The language we wish to download the music in, the artists or songs' time periods, and even the genre are the first things we need to determine. In this project, we are restricted to well-known English melodies without any consideration of duration. As a result, we gathered 764 songs with titles and artists from the internet that had catchy melodies. The format of this data was in CSV. Only the lyrics column from the dataset is required for this project.

B. Data Exploration

We used the Python script to investigate the dataset. In our dataset, there are 764 English songs with three columns. We'll start by counting the songs' characters, words, and lines. We discover that each song in the dataset has an average length of 1,405 characters, 318 words, and 46 lines. The lyrics will be generated by the model using the common terms it encounters during training. The word cloud may be used to view the terms that appear most frequently in the dataset. By putting together the word cloud we can observe that the most common terms in the dataset are Know, Love, Now, time, Say, Want, Take, and Come. It is seen in figure 1 below. Therefore, these words will typically be seen in Melody tunes. Because of this, it's possible that our model may produce lyrics with a love theme.

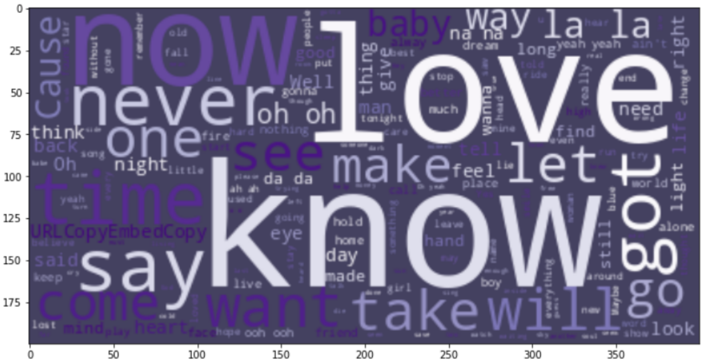

Fig. 1. Frequent Words in Dataset

As presented in Figure 2, when comparing the song durations of different artists, it can be seen that Ed Sheeran, a music composer, has the most lines and characters in his songs. The least amount of characters and lines belongs to composer John Denver. Character counts often range from 100 to 5000.

C. Data Preprocessing

In order to build a deep learning model for next character generation in lyrics, data preparation is a crucial step. Names of the artists, song titles, and song lyrics are all included in the data that was retrieved from the dataset. All of the lyrics in the dataset are contained in the lyrics column, which is where our model must be trained. We chose the music genre because it includes lengthy data sequences. Before fitting into

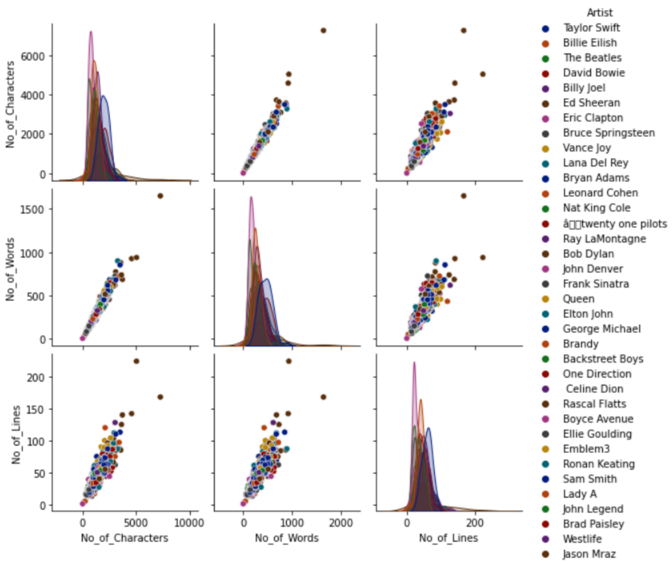

Fig. 2. Pairplot of Dataset by Artist

the models that we provide, the data has to be preprocessed. Preprocessing entails preparing the data for model training by cleaning and structuring the lyrics' data. This entails eliminating punctuation, lowercasing each word, and swapping out uncommon terms for tokens that are more widely used.

In order to train our model, we need a corpus of lyrics. Around 107,380 characters in all are available in the collection. When the number of unique characters in the corpus which is the vocabulary is counted, 98 or so are found to be present. However, there are 26 alphabets in the English language, and there are 0–9 numbers, so when we checked it further, we discovered that many foreign characters had infiltrated. Thus, we had to get rid of such undesirable characters. Around 98 distinct characters were present before preprocessing, while about 47 unique characters remained after preprocessing.

The new line character is inserted because the model has to understand when to insert it so that the last words of the song can rhyme. It will be necessary to separate the lyrics into smaller character sequences so that the model can be fed with them. These sequences, which are usually only a few characters long, are used to anticipate the sequence's subsequent characters. The characters and their indexes were then mapped using a dictionary. It assigns a number to each character in the vocabulary. Since it will be useful to encode and decode the information entering as an input and coming out as an output, the opposite is also done.Then, this cleaned data is divided into smaller, equal-length sequences. The labels are resized and normalized, and the targets are one-hot encoded. Separating the data into a training set and a validation set is crucial. The model will be trained using the training set, and its performance during training will be assessed using the validation set. 70% of the data are in the training set, 20% are in the validation set, and 10% are in the test set.

D. Deployment

The Streamlit framework makes it simple to develop and deploy interactive machine learning applications by enabling us to install a lyrics generator. With the models that we trained, we deployed our model using streamlit. The web application receives the seed sentence and the model that should be used in order to create lyrics. It also accepts the output, which is the amount of characters it must produce. The ability to update the lyrics generator in real-time as we make coding changes is one advantage of utilizing Streamlit for distribution. This implies that we don't need to keep re-deploying the application in order to test and improve the model. Overall, utilizing Streamlit to build a lyrics generator is a quick and easy procedure that enables us to quickly develop and launch interactive machine learning apps.

V. Models

In this project, we're going to use the seed as the beginning value to produce the lyrics. The first phrase or sentence of the song you wish to create is its seed. Producing lyrics is same to producing text. The sole distinction is the rhythm and underlying literary style that must be used. Lyrics are a group of words that have a meaningful relationship to one another. Therefore, sequential processing approaches are the best way to handle this issue. The first fundamental model created by Elman is the vanilla recurrent neural network (RNN). When compared to conventional neural networks, this model can function with a number of benefits. However, Vanishing Gradient is this model's flaw. Long Short-Term Memory (LSTM) and Gated Recurrent Units (GRU) are the next level architectural model that we are using to overcome this issue . These models don't experience vanishing or exploding gradients. Therefore, in this project, we are employing these components to develop the model. We proposing three models for this project; the specifics are shown below.

A. Model 1

LSTMs are a type of recurrent neural network (RNN) that are well-suited to tasks involving sequential data, such as language modeling. They are able to capture long-term dependencies in the data by using a special type of memory cell that can retain information for an extended period of time. This makes LSTMs particularly effective for tasks like lyrics generation, where the model needs to be able to take into account the context provided by the previous words in the sequence in order to generate coherent and meaningful lyrics. LSTMs are able to handle large amounts of data and can be trained on very long sequences, which is important for next character generation tasks where the model needs to be able to generate long sequences of characters. LSTMs are generally more robust to noise and outliers than other types of RNNs, which can be beneficial when working with real-world data that may contain errors or inconsistencies.

The first layer in the model architecture is the LSTM layer with 256 units and It is set to return sequences, which means that the output of this layer will be a sequence rather than a single tensor. The next layer is also the LSTM layer with 128 units. The final layer is a dense, fully-connected layer with 47 units and a softmax activation function, which means that it will output a probability distribution over the possible classes. The Adam optimizer is used with a learning rate of 0.001, and the model is compiled with categorical cross entropy as the loss function and accuracy as the metric. One advantage of having two LSTM layers is that it allows the model to learn and capture more complex patterns in the data compared to having just a single LSTM layer. This can potentially improve the performance of the model on the task it is being trained for. Additionally, using multiple LSTM layers can also allow the model to better handle long-term dependencies in the data, since each LSTM layer can learn to attend to different parts of the input sequence. This can be especially beneficial for tasks like maintaining the rhythm and style in the generating lyrics. In this model there are 467k trainable parameters in total.

B. Model 2

We felt LSTM is somewhat a complex model when compared with the Simple RNN, that's why we want to try any other model which is simpler than LSTM. Then we opted for GRU.

Gated recurrent units (GRUs) are a type of recurrent neural network (RNN) that are similar to long short-term memory (LSTM) models, but with a simpler structure. They can be a useful choice for next character generation tasks with accuracy as the evaluation metric because of their ability to capture long-term dependencies in the data. GRUs have a simpler structure than LSTM models, which can make them easier to implement and train, particularly when working with large datasets. GRUs can be trained faster than LSTM models because they have a simpler structure, which can be beneficial when working with large datasets or when time is limited.

The complexity of the model is reduced in GRU because two gates' input and forget functions are combined into one gate. Therefore, there will be two GRU layers in this strategy, each with 256 and 128 units. A dense layer with 47 neurons and a softmax activation function follows these layers. The prior model even had the same loss function. Additionally, there are less parameters than the prior model architecture's 353k. The difference between models' trainable parameters is nearly 110k.

C. Model 3

A bidirectional layer can enhance a text generation model's capacity to understand the context of the input text. In a traditional LSTM, the hidden state at each time step is only influenced by the previous hidden state and the current input. This means that the LSTM can only capture context from the past. A bidirectional layer processes the input sequence in two directions (forward and backward), which allows it to learn contextual information from both past and future words in the input sequence. This can be especially useful for tasks like language modeling, where the meaning of a word can depend on the words that come before and after it in the input sequence. Using a bidirectional layer can also help the model to learn long-term dependencies in the data, which can improve its ability to generate coherent and realistic text. Bidirectional LSTM models are more robust to noise and outliers than single LSTM models because they are able to capture dependencies in both directions, which can be beneficial when working with real-world data that may contain errors or inconsistencies. It is one of the reasons for using bidirectional layer in the model architecture. So, in this model we are combining the complex LSTM with GRU and also making these layers bidirectional. In this model we have 1033k parameters.

VI. Evaluation of Models

There are several different metrics that can be used to evaluate the performance of a model, depending on the specific goals and requirements of the task. We will finalize the evaluation metrics that can be applied to our models. In this project we are generating the next character by taking 100 input characters. So, we are predicting the 101th character in the sequence. We are using the accuracy and perplexity metric to evaluate our model.

Accuracy is a simple metric that measures the proportion of correct predictions made by the model. For lyrics generation using the next character, this could be calculated as the number of correct predictions divided by the total number of predictions. In the context of lyric generation, the model is given a sequence of words as input and is asked to generate a sequence of words that represents a coherent and meaningful set of lyrics. The accuracy of the model can be evaluated in several ways, depending on the specific task and the desired level of granularity.

Regardless of the specific approach used, it is important to carefully consider the metric used to evaluate the performance of the model, as it will influence the way the model is designed and the results that are obtained. In addition to accuracy, other metrics that may be useful for evaluating the performance of a model for lyrics generation include perplexity, which measures the likelihood of a sequence of words given the model. Perplexity is a measure of how well a model is able to predict the next character in a sequence, and is calculated as the exponentiated average of the negative log probability of the next characters, given the previous characters in the sequence. A lower perplexity score indicates that the model is more confident in its predictions and is therefore able to generate more coherent lyrics. Conversely, a higher perplexity score indicates that the model is less confident in its predictions and may generate less coherent lyrics.

A perplexity score of less than 2 is considered good, while a score of greater than 5 is considered poor. However, the optimal perplexity score will depend on the specific task and the quality of the training data. It is also worth noting that the perplexity score is highly sensitive to the length of the input sequence, so it may be necessary to compare perplexity scores for sequences of similar length in order to accurately compare the performance of different models. It is important to note that perplexity is one of the important metrics that can be used to evaluate the performance of a model for lyrics generation, and it should be considered in conjunction with other metrics such as accuracy.

VII. Results

In the below table 1, representing the performance of models with respective evaluation metrics. Model 3 which is with Bidirectional LSTM and GRU layers performed well when compared with the other two models. It is performed with the accuracy of 57.15% and also with the perplexity score of 4.3 which is not good as well as bad as mentioned earlier. The other two models are performed with less accuracy than the model 3 with 56.5% and 55.3% respectively. Even the perplexity scores of these models are more than the model 3 with the value 4.43 and 4.6 respectively.

TABLE I
Evalaution Metric Results

Evaluation Metric	**Models**		
	Model 1	***Model 2***	***Model 3***
Accuracy	56.5%	55.3 %	57.15%
Perplexity	4.43	4.6	4.3

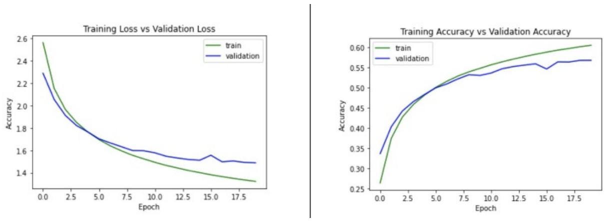

Fig. 3. Loss and Accuracy graphs of Model 1

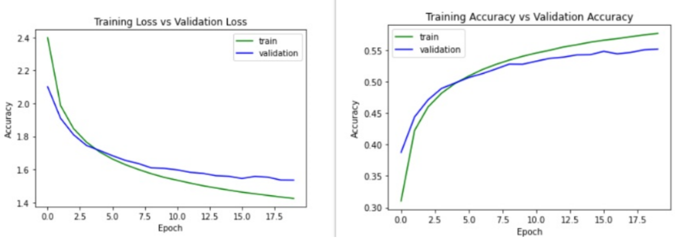

Fig. 4. Loss and Accuracy graphs of Model 2

The above are the graphs showing training loss vs validations loss and training accuracy and validation accuracy for each model. Each model is trained for 20 epochs. These graphs will help us to interpret the model performance in terms of overfitting or underfitting and learning speed. Model 3 shows slower learning than the other two models, but the other models are learning somewhat faster because of a simpler model than that model. Among all the models, model 3 outperformed the other models with accuracy and even with perplexity scores. If you see the graphs, you can see that models are overfitted but we stopped the training at the acceptable patience level.

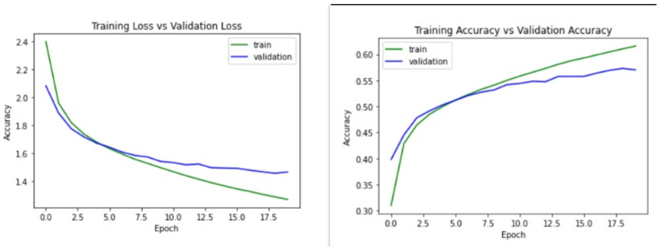

Fig. 5. Loss and Accuracy graphs of Model 3

VIII. Conclusion

From our experiments deep learning has shown to be a promising approach for lyrics generation, especially when using techniques such as long short-term memory (LSTM) networks. These models have the ability to capture long-range dependencies in language and can generate coherent and diverse lyrics that often exhibit the style and tone of the input data they were trained on. In our case model with bidirectional layers fo LSTM and GRU performed well than the other two models. When we are testing with some seed value it is giving better semantic and structural lines than the other two models. However, there are still limitations to the quality and coherence of the generated lyrics, and further research is needed to improve the performance of these models.

IX. Discussion and further improvements

Lyrics generation using deep learning is a challenging task that involves generating coherent and meaningful words and phrases to form a song. It involves the use of machine learning algorithms and techniques, such as artificial neural networks, to process and analyze large amounts of data to generate original lyrics. One of the key challenges in using deep learning for lyrics generation is the quality of the generated lyrics. While some studies have shown that deep learning models can generate high-quality lyrics, there is still room for improvement in this area.

One potential approach to improving the quality of generated lyrics is to use more diverse and comprehensive training datasets. The quality of the generated lyrics is largely dependent on the quality and variety of the data that the model is trained on. By using a larger and more diverse dataset, the model can learn a wider range of patterns and structures, which can improve the quality of the generated lyrics.

Another potential improvement is to incorporate other types of data into the training process. In addition to lyrics, the model could be trained on other types of text, such as poetry or prose, in order to learn a broader range of language structures and styles. This could help the model generate more diverse and creative lyrics.

In addition to the quality of the generated lyrics, there is also the question of how the generated lyrics can be used in practice. One potential application is to use the generated lyrics as a starting point for human songwriters. The songwriter could use the generated lyrics as a source of inspiration, and then modify and adapt the lyrics to fit their own creative vision. This could help songwriters generate new ideas and overcome creative blocks.

Training the model on lyrics in certain language and so that the model generates lyrics in that particular language. Including songs of specific artists and genres and the model tries to mimic their style of writing. Trying to adopt for a transformer based learning methodology so that the song makes much meaning. Creating a music generator application which tries to generate music in sync with the lyrics present.

Overall, there are many potential avenues for further improvements in the use of deep learning for lyrics generation. By continuing to develop and refine the underlying technology, as well as exploring new applications and uses, deep learning has the potential to revolutionize the way we create music.

References

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

[2] Fernandez, Aaron Carl T., Ken Jon M. Tarnate, and Madhavi Devaraj. ”Deep Rapping: Character Level Neural Models for Automated Rap Lyrics Composition.” International Journal of Innovative Technology and Exploring Engineering (IJITEE) ISSN (2018): 2278-3075.

[3] Li, Anqi, Davin Raiha, and Kenneth W. Shotts. ”Propaganda, alternative media, and accountability in fragile democracies.” The Journal of Politics 84, no. 2 (2022): 1214-1219.

[4] Zhang, H., Liu, X., Fan, Y. (2020). Towards more human-like song lyrics generation with deep learning. In International Conference on Neural Information Processing (pp. 856-866). Springer, Cham.