Deep Learning – Report

Introduction

This project focuses on building a recurrent neural network (RNN) to generate song lyrics based on provided melodies. The goal is to enhance understanding of deep learning applications in creative tasks, such as automatic lyric generation, and to evaluate various approaches for incorporating musical elements into the model.

This report describes the dataset used, methods, experiments, and results.

*In the final submission, we have attached a notebook (.ipynb) and a Python file (.py). Initially, we ran offline (sbatch) with several different training configurations as arguments and saved the models. Then, we loaded the trained models in the notebook and conducted the entire evaluation.

Dataset

The dataset for this project consists of two main components: MIDI files and lyrics. The MIDI files contain melodies from various songs, storing intricate musical details such as notes, timing, and instrument information. These files provide a rich source of data that allows the neural network to understand the musical context of the lyrics.

Additionally, a CSV file contains the singer name, song name and the corresponding lyrics for each melody, providing textual data for training the model. The dataset includes 600 songs for training and 5 songs for testing. Each word in the lyrics is represented using Word2Vec embeddings, capturing semantic meanings and relationships between words. The combination of MIDI and lyrical data enables the network to learn the interplay between music and lyrics, facilitating the generation of coherent and contextually appropriate lyrics based on the given melody.

The dataset had problems with mismatched song names between the CSV and MIDI files, and some MIDI files had audio issues. We fixed some of these problems and saved some songs, but we had to discard some due to unresolved issues. Screen shot of the problematic midi files:

```
Failed to load midi_files/beastie_boys_-_girls.mid: Could not decode key with 1 flats and mode 255
Failed to load midi_files/billy_joel_-_movin'_out.mid: data byte must be in range 0..127
Failed to load midi_files/billy_joel_-_pressure.mid: data byte must be in range 0..127
Failed to load midi_files/dan_fogelberg_-_leader_of_the_band.mid: Could not decode key with 4 flats and mode 255
Failed to load midi_files/brian_mcknight_-_on_the_down_low.mid:
Failed to load midi_files/aaron_neville_-_tell_it_like_it_is.mid: data byte must be in range 0..127
```

Pre-Processing

Text Lowercasing

All lyrics text was converted to lowercase to maintain consistency and avoid discrepancies between the same words in different cases.

Punctuation Removal

Punctuations were systematically removed from the text, ensuring the text only contained alphanumeric characters and spaces.

Spelling Correction

- A spelling correction function was utilized to fix misspelled words. This involved a custom function that checks if each word exists in the GloVe token-to-index dictionary (glove.stoi).
- Words not found in the dictionary were corrected using a SpellChecker. Corrected words that appear in the GloVe dictionary replaced the original ones in the text.

Text Cleaning

- Specific text contractions and colloquial expressions were expanded to their formal equivalents (e.g., converting "won't" to "will not").
- General contractions involving possessive cases and negations were also standardized (e.g., removing apostrophes and expanding "n't" to "not").

Integration of GloVe Embeddings

- We utilize GloVe embeddings for text representation, specifically selecting the "6B" model with 300-dimensional vectors from the torchtext.vocab library. These embeddings, pretrained on a corpus of 6 billion words, offer extensive linguistic information that facilitates understanding of text semantics and syntax, which is crucial for natural language processing tasks.

Tokenization and Index Mapping

- After cleaning and spelling correction, the text was tokenized. Each unique word, confirmed to be in the GloVe dictionary post-correction, was then mapped to a unique index. This mapping was stored in word2index and its inverse in index2word.
- Custom Index Mapping: Instead of using the original indices from the GloVe dictionary, we assign new indices to the words. This approach is taken to minimize the size of the index vector, thereby reducing the memory footprint and computational requirements of our model. By focusing only on words relevant to our training dataset, we significantly decrease the operational overhead associated with managing a large vocabulary.
- The token "eos" (end of song) was added to the end of each processed lyrics entry to signify the end of the song.

Processing Melody Data from MIDI Files

In our project, we explore the integration of melody data extracted from MIDI files to enhance the model's understanding of the relationship between music and lyrics. Here's how we approached the feature extraction from MIDI files, employing two different methodologies and a third option that combines elements of both:

MIDI File Processing Framework

The melody data is sourced from MIDI files corresponding to each song in our dataset. Our custom LyricsMelodyDataset class is designed to handle the preprocessing and extraction of relevant features from these MIDI files, ensuring that each song's lyrical content is synchronized with its musical composition.

In our dataset construction, we focus on two key parameters: frames_per_word and avg_duration_of_word.

1. **frames_per_word** determines the number of frames representing each word's musical segment in the MIDI data, influencing the detail level of the musical analysis per word.
2. **avg_duration_of_word** calculates the average time duration each word occupies within a song. This metric is derived during training from the total song durations and the number of words, ensuring proportional musical representation for each word. For validation and testing, this average is directly taken from the training data to maintain consistent and fair evaluation standards across different dataset splits.

Feature Extraction Approaches

1. **Melody-Based Features**:
    - Fixed Size Segments: We generate fixed-size segments of melody data from the MIDI files. Each segment corresponds to a fixed number of frames per word in the lyrics, ensuring a consistent representation across all songs. This method focuses on capturing the raw piano roll output from the MIDI data, segmenting it according to the predetermined frames per word.

2. **Instrument-Based Features**:
    - **Detailed Instrument Analysis**: This approach dives deeper into the nuances of each MIDI track by examining various characteristics of the musical composition during the duration of each word. Features such as the number of instruments active, the presence of drum tracks, average pitch, velocity of notes, and changes in beat within each segment are calculated. This method provides a rich set of features that represent the musical dynamics more comprehensively.

3. **Combined Features**:
    - **Hybrid Feature Set**: Recognizing the strengths of both the melody-based and instrument-based approaches, we implemented a third option that merges

these methodologies. By concatenating the features extracted from both the fixed size melody segments and the detailed instrument analysis, we create a hybrid feature set that leverages the benefits of both approaches. This combination aims to provide a more holistic representation of the musical elements corresponding to each lyrical segment.

Usage in Model

This processed MIDI data is then integrated into our training and validation datasets. The features serve as inputs to our neural network models, where they are used alongside word embeddings from the lyrics to predict various aspects of song composition and lyrical alignment.

Architecture of the Lyrics Generator Model

Our model, LyricsGenerator, is structured as a neural network module that integrates both lyrical and musical data for generating lyrics. Here’s how we designed its architecture:

Core Components

- **LSTM Layers**: The backbone of the model is formed by LSTM units. This choice is motivated by the LSTM's capability to handle sequences, such as lyrics and time-based musical features. We configure the LSTM with a combination of the embeddings dimension (embedding_dim) from the lyrics and the features dimension (feature_dim) from the music. The LSTM outputs are passed through multiple layers (controlled by num_layers), and dropout (dropout) is applied between these layers to prevent overfitting, especially when the number of layers exceeds one. Notably, our LSTM is unidirectional and functions solely as a decoder in the architecture, focusing on generating lyrics sequentially without backward input integration.

- **Fully Connected Layer**: Post-LSTM, a linear transformation is applied via a fully connected layer (fc), which maps the LSTM's output to the desired output dimension (output_dim). This layer is crucial for shaping the LSTM output into a format suitable for our specific lyrical generation task.

- **Dropout**: An additional dropout layer is included post-LSTM to further aid in regularizing the model, enhancing its ability to generalize by reducing dependency on any particular set of neurons.

Forward Pass

- **Input Preparation**: During the forward pass, music features and lyrics embeddings are first flattened and concatenated. This combined input is crucial as it merges textual and musical contexts, providing a rich feature set for the LSTM to process.

- **Processing Sequence**: The combined input is fed into the LSTM. The output from the LSTM undergoes dropout to mitigate overfitting, followed by the fully connected layer that produces the final output of the model.

Model Configuration

- **embedding_dim**: This parameter is set based on the dimensionality of the GloVe embeddings we use, which is 300.

- **feature_dim**: The feature dimension corresponds to the features extracted from the MIDI files. If using the combined approach (melody and instrumental features), this dimension would be the sum of the dimensions from each feature set. For each frames_per_word the dimension of Melody-Based Features is 128 and of Instrument-Based Features is 134.

- **Output Dimension (output_dim)**: This dimension is 6861 as the vocabulary size of the lyrics, which is determined by the word2index mapping built during data preprocessing.

Words Generation

In our model's implementation for generating lyrics, we incorporate a Top-k Sampling approach to select the next word in the sequence, ensuring that our text generation mechanism remains probabilistic rather than deterministic. In our case we chose k as 5. This method enhances the creativity and variation in the generated lyrics. In addition, we set the max length of the generated lyrics as 300 words per song.

Evaluation

Given the large scale of our model, which required substantial time to train, we limited our experimentation to a few configurations. Specifically, we selected two distinct frames_per_word settings: 1 and 10. These configurations were trained using three different approaches previously outlined in this report: Melody-Based Features, Instrument-Based Features and Combined Features. Additionally, the data was partitioned into training and validation sets with a 99:1 ratio, allowing for consistent model evaluation while maximizing the data available for training.

After each training epoch, we measure performance metrics on the validation set to monitor the model's progress and adjust parameters if necessary. To ensure that we can recover and analyze the model at various stages of training, we save a checkpoint of the model every 100 epochs.

Experiments Configurations

- **Number of Layers (num_layers)**: We utilize a 2-layer approach, to enhance the model's ability to learn complex patterns in the data.
- **Dropout (dropout)**: Set at 0.3 to help prevent overfitting by randomly omitting units from the learning process during training.
- **Hidden Dimension (hidden_dim)**: Typically configured based on model complexity and dataset size. We set it to 256, which allows the LSTM to develop a robust internal representation of the combined inputs.
- **Learning Rate (lr):** Set to 0.001.
- **Epochs:** 400.

The model parameters are updated using the Adam optimizer, chosen for its effectiveness in handling sparse gradients on large datasets like ours. Additionally, we incorporate a Cosine Annealing Learning Rate Scheduler, which gradually reduces the learning rate following a cosine curve from its initial value to near zero by the end of the training. This scheduler is particularly beneficial as it helps in avoiding local minima and ensures smoother convergence, which is crucial given the complexity and length of the training process.

Loss Calculation Methods

The loss function used in our project combines multiple elements to evaluate the quality of the generated text. Here's a breakdown of the components of the loss function:

CrossEntropyLoss (logits_loss): This is a standard loss function for classification tasks, which measures the difference between the predicted probabilities (logits) and the actual class labels (words in this case).

num_of_lines_loss: This part of the loss function is designed to capture two aspects of the song structure:

- Line Count Difference: This calculates the absolute difference in the number of lines between the generated song and the target song. It ensures that the structure in terms of the number of lyrical lines aligns with what is expected.

- Average Words Per Line Difference: It computes the absolute difference in the average number of words per line between the generated text and the target text. This checks whether the lyrical density per line matches that of the target, reflecting how well the model captures the lyrical density driven by the musical input.

Combined Loss Calculation (loss):

The total loss is a weighted sum of the logits_loss and num_of_lines_loss. The weight of the line structure part (weight_of_lines) can be adjusted to emphasize the importance of getting the structure right relative to just getting the words right. We set weight_of_lines to 0.01. The final formula of the loss function is:

$$Loss = \text{CrossEntropyLoss} + \text{weight\_of\_lines} * \text{num\_of\_lines\_loss}$$

Evaluation Metrics

To effectively evaluate our model's ability to generate song lyrics, we use several key metrics on both the validation and test sets. This approach provides a thorough analysis of how well our model performs in creating lyrics that are both relevant and contextually accurate. Here's a breakdown of each metric and why it was chosen:

- **Cosine Similarity using text encoder (DeBERTa):**
  We use cosine similarity to measure the semantic closeness between the generated lyrics and the original lyrics. This metric utilizes embeddings from a DeBERTa model, giving us an understanding of the underlying meanings and themes in the lyrics.

- **Jaccard similarity:**
  Jaccard similarity assesses the overlap of unique words between the generated and original songs lyrics. This metric is straightforward and effective in measuring the exact word matches. It evaluates the model's ability to use correct and expected words within the lyrics, reflecting the model's lexical accuracy and appropriateness in lyric generation.

- **BLEU Score:**
  The BLEU score is widely used in evaluating text generation tasks like machine translation, making it suitable for assessing lyrical generation. It measures how many words and phrases in the generated lyrics match those in the reference lyrics. This score is particularly challenging for lyric generation tasks because it strictly assesses exact matches in word and phrase sequences, which may not fully capture the creative and varied linguistic expressions typical in song lyrics.

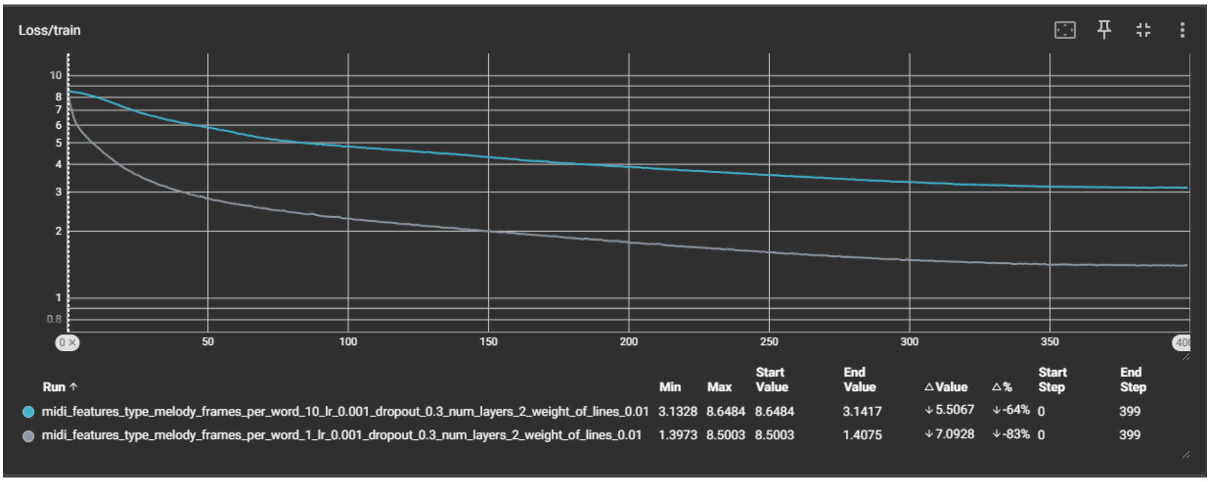
Results

In the results section, we present a detailed analysis of our model's performance across training, validation, and testing phases. Initially, we experimented with two different settings for the frames_per_word hyperparameter: 1 and 10 only on the Melody-Based Features approach.

Hyperparameter Analysis - frames_per_word

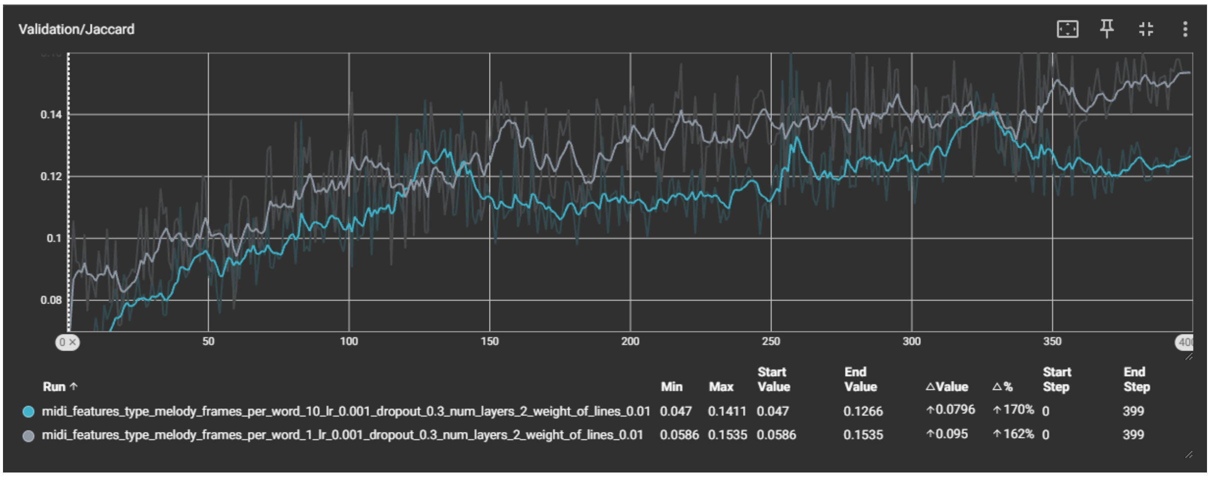
Training Results

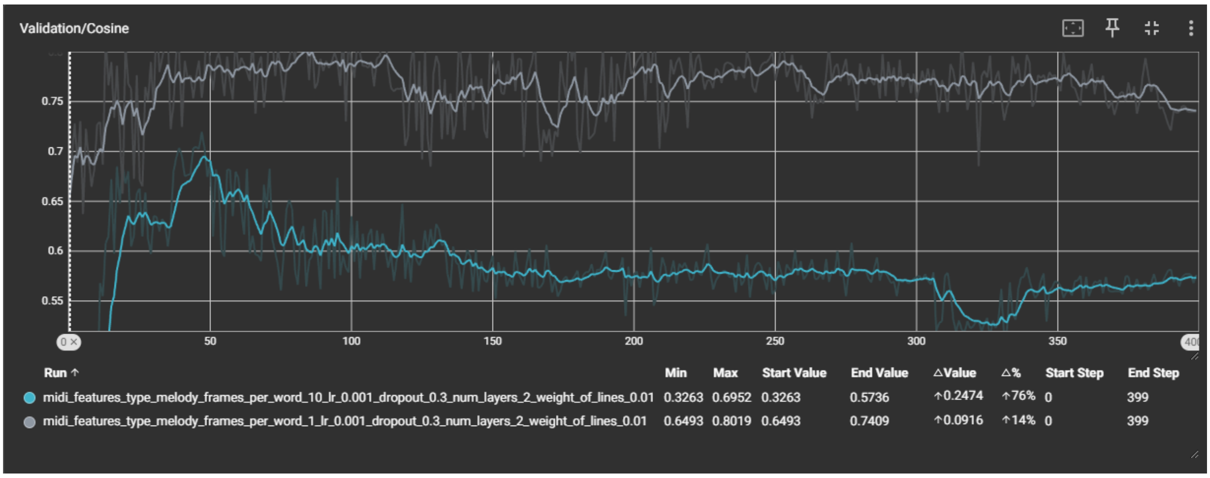
In this subsection, we detail the model's performance during the training phase, supported by loss graphs generated from TensorBoard. These graphs illustrate the model's loss trends over 400 epochs, providing insights into the learning process and the effectiveness of the learning rate adjustments made by the scheduler.

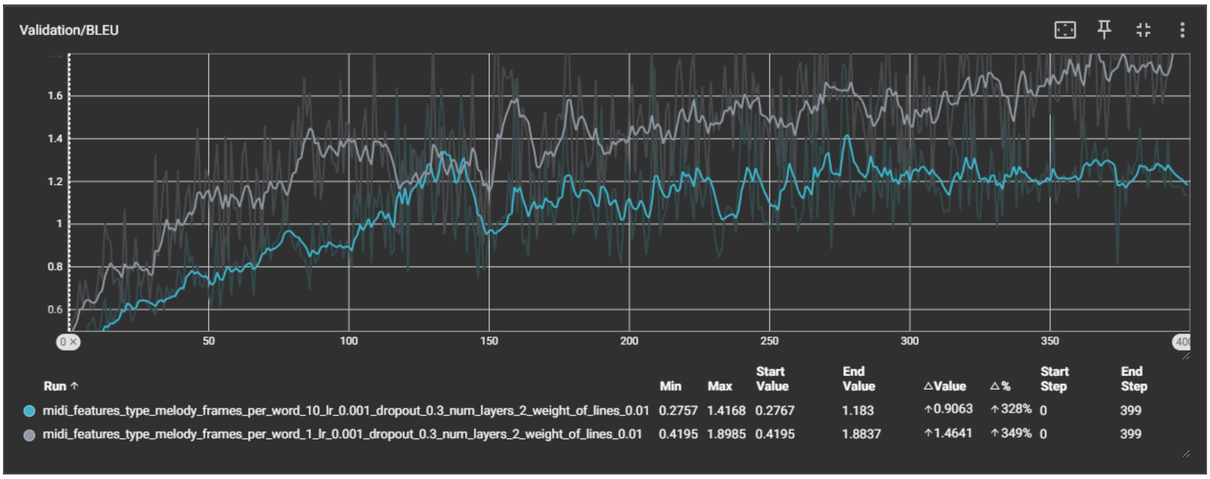


Validation Results

Here, we explore the performance of our model on the validation dataset using the metrics previously described: Cosine Similarity, Jaccard Similarity, and BLEU Score. This analysis helps validate the consistency and reliability of the model before it is tested on unseen data.







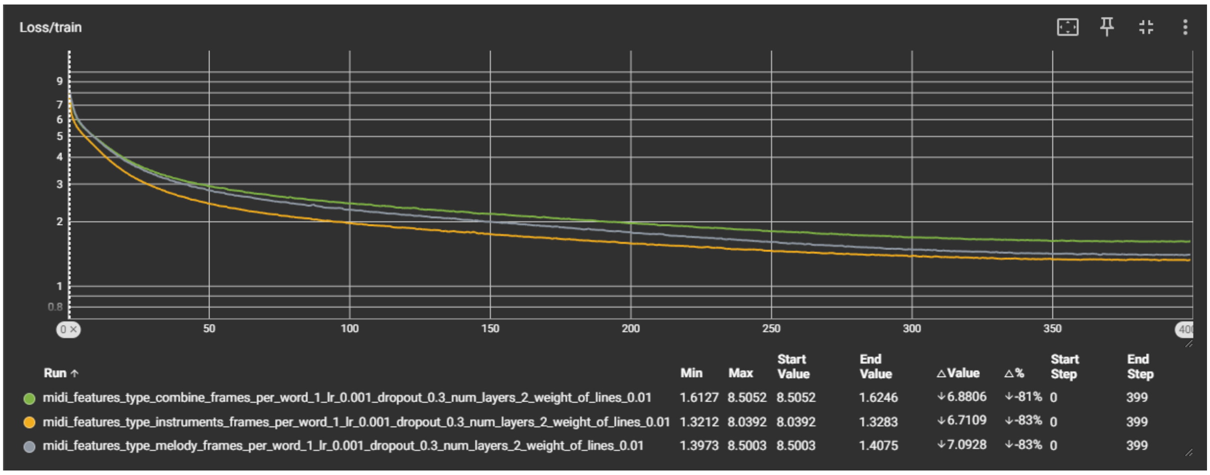
The comparative analysis indicated that a frames_per_word setting of 1 yielded superior results in terms of both evaluation metrics on the validation set and loss on the training set. Therefore, the subsequent parts of this section will concentrate exclusively on the results obtained with this parameter setting.

From the analysis of training loss and validation metrics, it is evident that the model continues to improve with further training. Consequently, we have decided to utilize the checkpointed models from the 400-epoch mark for the remainder of this report, as they demonstrate the most robust performance and learning progression up to this stage.

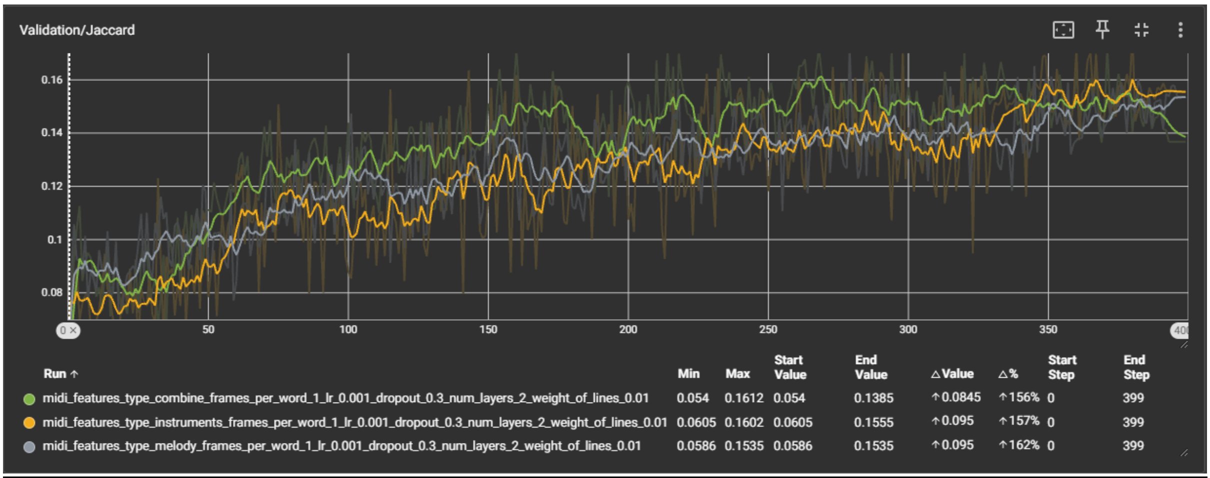
Analysis of Feature Extraction Approaches

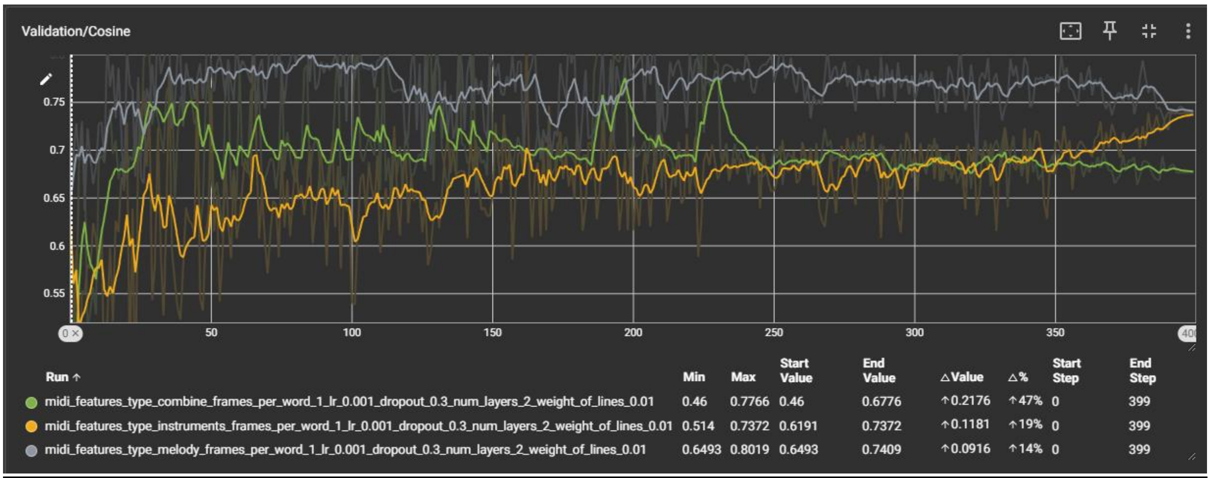
Our study evaluates three distinct MIDI data feature extraction strategies to enhance lyric generation: Melody-Based Features, Instrument-Based Features, and Combined Features. By assessing these approaches, we aim to identify which offers the optimal balance of complexity and performance, informing future model optimizations and feature engineering efforts.

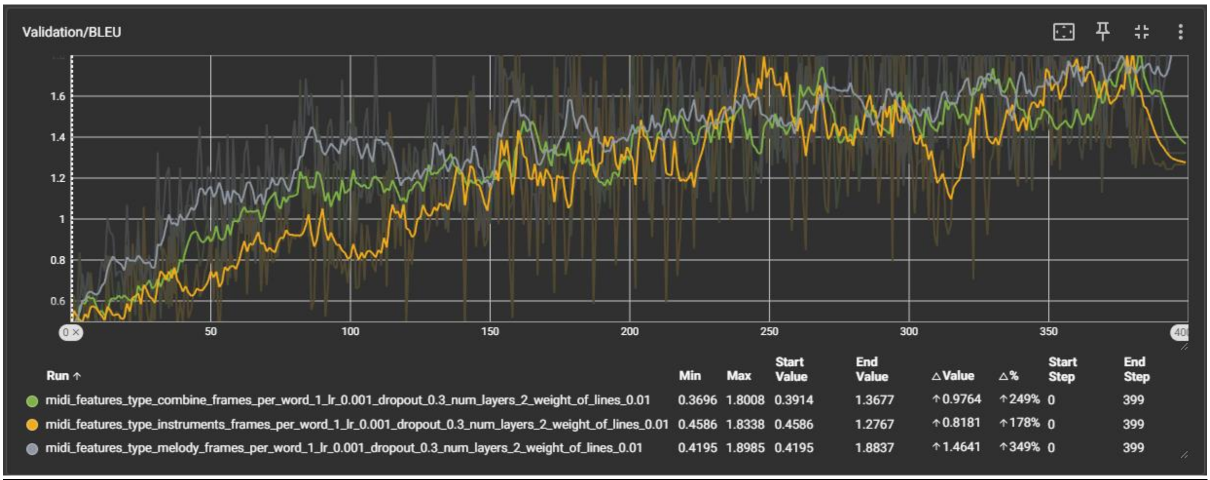
Training Performance Analysis Across Feature Extraction Approaches



Validation Performance Comparison







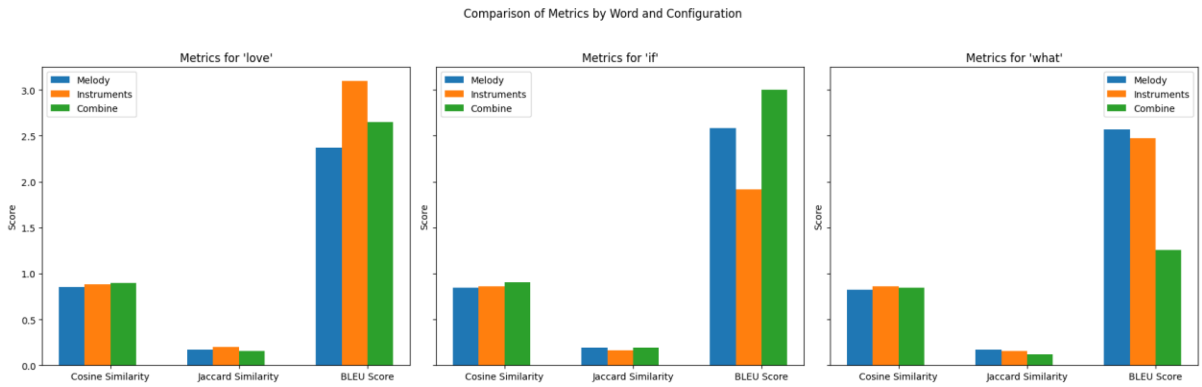
Based on the training and validation results, we observe that no single feature extraction approach—Melody-Based, Instrument-Based, or Combined Features—significantly outperforms the others. The metrics from these evaluations do not provide a clear distinction to determine which method is superior. This outcome suggests that the impact of these musical feature extraction techniques on lyric generation might not be as substantial as the lyrics themselves. Further investigations may be needed to explore other factors that could have a more pronounced influence on the model's performance.

Test Results

The final subsection presents the model's performance on the test set (5 songs). This is crucial as it reflects the model's ability to generalize to new, unseen data, effectively demonstrating its practical utility in generating song lyrics.

For testing, we chose 3 different initial words as first input to the models: love, if, what.

Here are the plots comparing the three metrics—Mean Cosine Similarity, Mean Jaccard Similarity, and Mean BLEU Score for each initial word across the different configurations (Melody, Instruments, Combine).



The charts show that using different starting words slightly changes the results, but not by a lot. This small difference is likely because our model doesn't always pick words in the same way every time—it chooses based on chances, which can vary. So, even if we start with the same word, the lyrics that follow might be different each time. This shows that our model is creating varied lyrics, adjusting to different starting points and music.

For the testing phase of our model, we introduced two additional metrics to gain deeper insights into the generated lyrics: sentiment analysis and genre classification.

Sentiment Analysis

This metric assesses the emotional content of the lyrics. We utilize a sentiment classifier from the `transformers` library, specifically the model "michellejieli/emotion_text_classifier". This classifier provides scores for different emotions, allowing us to compare the emotional spectrum of the generated lyrics against the original.

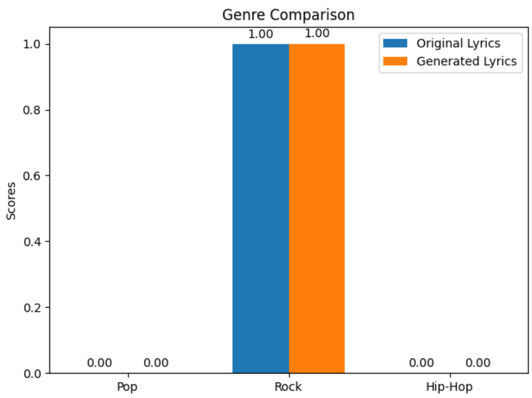
Genre Classification

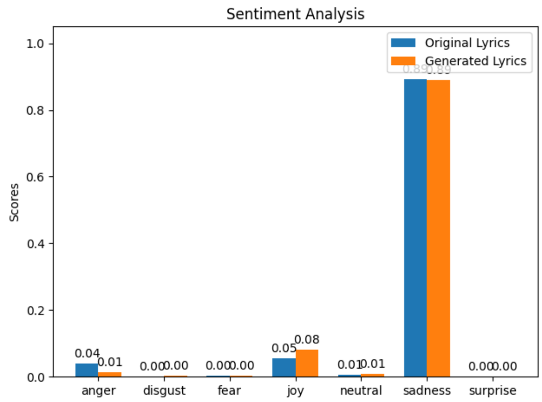
This classifier predicts the musical genre of the lyrics, using the model "Veucci/lyrics-to-genre" from the same library. It gives us an idea of how well the generated lyrics align with genre-specific language and themes. Similar to sentiment analysis, the genre classifier returns scores for various genres .

Both metrics are visualized using bar graphs that display and compare the scores of the original and generated lyrics, providing a clear visual representation of how well the generated content matches the original in terms of emotional depth and genre specificity.

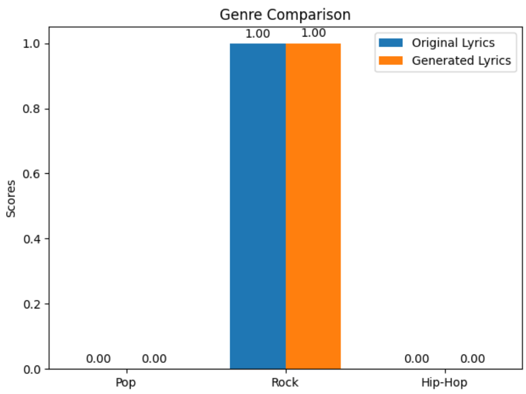
Here is a one example of the same song generated with the same first initial word – "love", on all the 3 approaches (Rest of the songs are attaches in the notebook):

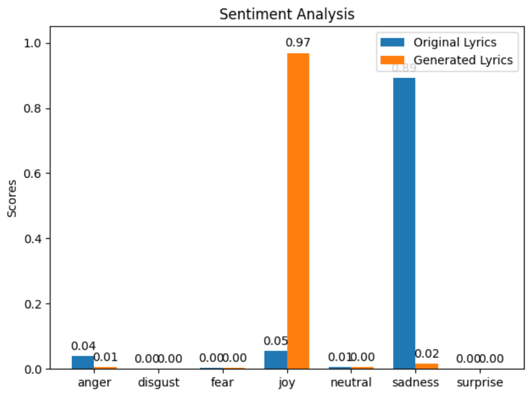
Melody Features - "Honesty" of Billy Joel:



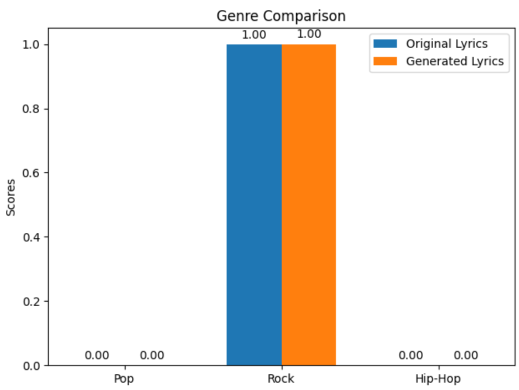


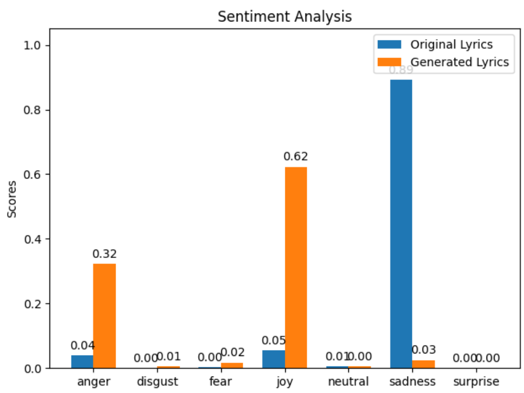
Instruments Features - "Honesty" of Billy Joel:





Combined Features - "Honesty" of Billy Joel:





Songs Generation

Here is an example of the same song generated with 3 different initial words – "love", "if", "what" on all the 3 approaches (Rest of the songs are attaches in the notebook). The order of the songs is: original, melody, instruments, combine.

"Honesty" of Billy Joel:

```
If you search for tenderness
It isn't hard to find
You can have the love you need to live
But if you look for truthfulness
You might just as well be blind
It always seems to be so hard to give
Honesty is such a lonely word
Everyone is so untrue
Honesty is hardly ever heard
And mostly what i need from you
I can always find someone
To say they sympathize
If i wear my heart out on my sleeve
But i don't want some pretty face
To tell me pretty lies
All i want is someone to believe
Honesty is such a lonely word
Everyone is so untrue
Honesty is hardly ever heard
And mostly what i need from you
I can find a lover
I can find a friend
I can have security until the bitter end
Anyone can comfort me
With promises again
I know i know
When i'm deep inside of me
Don't be too concerned
I won't ask for nothin' while i'm gone
But when i want sincerity
Tell me where else can i turn
'cause you're the one i depend upon
Honesty is such a lonely word
Everyone is so untrue
Honesty is hardly ever heard
And mostly what i need from you
```

```
Love look at the two of our love
Strangers in the dark
This is the end of the world
The time of the spring
The stars is calling
Are the night in the night
The day of love
A little one life
We can be a star for the last
And there of the world
You are the one dance
When you are the same
You are the perfect girl
And i can not see it
When you are not a friend
When you are a fool in a man girl
I will be a fool
And who you are coming at me
You are gonna be a fool
When you are not not you
Do you really want to know
You are never to do it
When you are not a fool
When you are coming on the stranger
You are never so much
When you are not forgotten
Why are you
You are not a fool
Who will be cool
When we are in the state
You belong
In the beginning of the cat
```

```
Love look at me i am the one
I am thinking of a love i have known
I have been searching for the daughter
I am trying to get to tell you
I say you have been taken a thousand years
And i know i would like to make it with
I am a woman in love
And i do not know what i have found
I am not the woman that you can not see
I will be your friend figure
And i am thinking bout a little prayer
I will find my love again
And i know that i will be there
I will be the one
I will be your friend figure
I will be the one
I will be your friend figure
And i can feel your eyes through my heart
And i will find my life away
Cause i am not there i can go
I will give you all my life every day
And i will be the one
I will be the one
I will be the one
I will be the one
I will be the one
I will be the one
I will be the one
I will be the one
I will be the one
I will be the one
I will be the light
When i feel the magic floating in love
And i can not believe it
```

```
Love look at the world
I think you have to be a guy
I have been a fool of a child
I was a little who i have known
And i have been been a fool for me
I have been a fool for lesser and i have been strong of my heart
I have known to make you to know
And i will be waiting for you
I am in a fool for you
I am a little one two you
I am gonna get it to my heart
I have made a lot that i am in love with me
I know i have seen a lot for a star
I am a little who i am gonna break my mind
I have known to make you to be a star
I will be a one to keep my own
I will be the one
I will be your preacher
I will be your preacher
I will be your father
I will be your preacher
I will be your preacher
I am a little one
I am gonna make a world of the fight
I will be the one
I will be the one
I will be the one i will be a fool
I will be the one i will be a one
I will be a fool
I will be the one i have to be
To the one for you
```

```
If you want to get it to do
You have got a lot of me
You would make it take to get it
You can dust it off and try it on
It is not you just let
We think about the things that is a fool
You are never gonna get it
But you are never gonna get it
Do not you know that you are going
You are never gonna get it
But you are gonna get it
But you are better for me
Do not you know you are not a fool
Let me get around and you
Do you want to let me go
You can never let it
Open you can never leave
Cause if you go
Just a way of your heart
So can not you know
You can never let me
Cause you can take it all as try
Get around and try it on the floor
If you want to dance
Open your eyes
There is a woman
And if we are in the way
You are in love with me
And if you put your heart
And you will see you
```

```
If you want it here it is come and get it
Mm mm mm mm make your mind up fast
If you want it any time i can give it
But you would better hurry cause it may not last
Did you hear what you do
If you want it any way i can not get it
Cause you would not have to do it
Cause you would not hide my colour
And i will not have to play it off
And i am gonna be a fool
I am gonna fight you
I am gonna fly on down
I wanna get up my mind
Do not know what you do
I do not know how to do
I do not want to be your clown game
I do not wanna be your clown now
Cause i can not help myself
I do not wanna be your clown now
I just can not seem to get it right
I do not wanna be your clown
I will not be your father
I will help you now
I do not want to be your clown
Just lay to keep on me
I just want to be with you
I want you for you
I know that you will not be there
I will be your father
I will help you all i will not lie
I will survive
I will not be your arms on your side
```

```
If you want to be a chance to you
You do not want to be a fool
You would not say it is a game
That you are going to get a dream
I am gonna get it to you
You would be coming to be a fool to keep you
You are gonna make a chance to make you know
I will be your father
I am gonna get it to be with you
I am gonna be with you
I am gonna be with you
I am gonna be strong
I am gonna be strong
I am gonna be strong
I am gonna be strong
And i am gonna be strong
I am gonna make you and i am gonna break
And i am not gonna give you right
I will be your arms
If you say that i would not resist
I will be the one you would be
I will be your arms
I need to be your arms
I would be your love you can not let me go
I will love you better than that
I am gonna love you
Over the reason i want to know
I am gonna make it all i do
I am gonna make it right
I am gonna love you all i am with you
I do not want to fade away
I do not want to fade away
```

```
What kind of those lonely life
When i have been there gonna be
I am in love with you
And i am thinking you are gonna go
I am in love
I am sorry that you are mine
I am not you are believing
You are no one
Who caught me in the way
You are my voice
You gave me high
To make you happy
And i am in your eyes
I drove in night
Crept out
You are my obsession
Who was a quiet thing to hide
I am in love with me
I am a girl in you are all
And i see you could not see
I am not a fool
You are just
You are my obsession
I bought my soul
Who you are mine
I am a vision
You are always for me
And i am not a fool
Was not a problem it is been hard to hurt
I am feeling in you
I am blessed
For breaking your heart
And i am your eyes
Now i am your eyes
```

```
What child is this who laid the rest
The moon is sleeping and the sun is on
Oh wo ho the king is not
The moon is on the wind
And the moon is on and the sun is on
Oh what a little boy for a life
In the church of the night
In the church of the people of the world
Of the moon of blue blue eyed
He is in the earth of the night
He is in the eye of the night
He is just a hired that
He is gonna get caught
He is just the same of the people
He is just a little woman
He is gonna get out of the night
He is fighting and biting and riding on his horse
He is going and the sun is right
He is gonna be strong and he is a little bit
With a voice he is a lot of one
And he is going for speed
He is going for speed
He is going for speed
He is going for speed
He is going for speed
He is going for speed
He is going for speed
He is going for speed
He is going for speed
It is not the same
He is going for speed
He is going for speed
He is going for speed
He is going for speed
He is going for life
```

```
What child is a little life
And what it was not a friend
Who was a place of the man
He is a little life for the man
He was fighting the light of a winter
He is a dream of a hand
He is not a fool of a bottle
But he is not a lovely time
With the country of the poison life
In the country the past
He is going to be a dream of a hand
He is a man of a hand
He knows that he was not found
To be the one of the country
He is a place of the country
He read the swingers he is in her eye
He says that he is not
He is the son of a preacher
He was a sister of the fight
He was the kind of the queen
He knows that he was
But he is all a drag
He is the one that he was
He is a preacher boy
He is the one of the fight
He read a dream that the life
He is the queen and the law same
He is a little man for a preacher man
In the country of the country
He is a place that the big world
He will not tell the world of the fight
He will bring him the place to fight
And the sun moon with the
```

Conclusions

In this project, we developed a lyrics generation model capable of producing song lyrics that are both linguistically coherent and thematically aligned with specific musical inputs. By employing advanced natural language processing techniques and integrating music feature extraction methods, our model demonstrated its ability to generate text that reflects the underlying musical context. Although our evaluation utilized multiple metrics such as Jaccard, Cosine Similarity, and BLEU scores for linguistic accuracy, along with tests for sentiment and genre appropriateness, we think that the BLEU score may not be the most suitable metric for this task due to its strict reliance on exact matches and order. Furthermore, while the initial words impact the generated lyrics, the results vary, even when regenerating lyrics for the same song with the same initial words. This variability underscores the model's capability to capture both the essence and the structure of music-influenced lyrical content, but also highlights the inherent challenges in achieving consistent outcomes.

Challenges

Several challenges arose during the project, impacting various aspects of model development and performance evaluation:

1. Model Complexity and Training Time: The complexity of the model, combined with the need for extensive training data, resulted in significant computational demands and long training times. Additionally, the training dataset was relatively small, which posed significant challenges in training a robust model.

2. New and Complex Architecture: Working with a complex architecture such as LSTM for the first time introduced implementation challenges.

3. Variability in Music and Lyrics Alignment: The inherent variability in how music can be interpreted into lyrics posed a challenge, as different musical features can lead to vastly different lyrical outcomes, making consistent evaluation difficult.

Future Work

To build on the current project and address the encountered challenges, several areas of future work are proposed:

1. Exploring Alternative Architectures: Investigating other neural network architectures like GRU, Transformers, or encoder-decoder models could provide insights into different aspects of model performance and efficiency. These architectures might offer improvements in handling the complexities of lyric generation.

2. Enhanced Hyperparameter Optimization**: With more time and computational resources, conducting thorough hyperparameter optimization could improve the model's performance.

3 .Improved Metrics for Creativity: Developing or integrating new metrics that better capture the creative aspects of lyrics could help in fine-tuning the balance between creativity and accuracy. This would ensure that the generated lyrics not only conform to musical standards but also exhibit artistic flair.

4. Interactive Lyric Generation: Implementing an interactive system where users can provide real-time feedback or direction could refine the model's outputs and increase user engagement. Such a system would allow for more personalized and contextually appropriate lyrics.

5. Multilingual Support: Expanding the model's capabilities to generate lyrics in multiple languages would enhance its accessibility and applicability in diverse musical contexts globally.