This project examines the potential of generating song lyrics using LSTM models. By training on a dataset that includes both lyrics and melodies, the goal is to develop a system capable of creating lyrics based on given melodies. The melodies, stored in MIDI format, provide a rich source of musical data, which the model learns to process alongside textual information.

Data Preparation:
Effective data preparation is crucial to ensure both the lyrics and melodies are structured and optimized for machine learning tasks. The following outlines the preprocessing steps for each component:

Lyrics Processing:

1. **Cleaning and Special Character Removal:**
   Special characters and non-essential punctuation, such as exclamation marks or commas, are removed to reduce noise and maintain focus on meaningful content.
2. **Tokenization and Case Standardization:**
   The lyrics are split into smaller units, or tokens, such as words or symbols, and converted to lowercase. This helps maintain consistency and reduces vocabulary size.
3. **Contractions Expansion:**
   Abbreviations and shortened forms like "isn't" are expanded into full expressions, such as "is not," to standardize the text for processing.
4. **Word Embeddings:**
   Pre-trained embeddings, such as Word2Vec, are used to convert words into dense vector representations. These embeddings capture the semantic relationships between words, leveraging existing knowledge from large text corpora.

Melody Processing (MIDI Data):

1. **Feature Extraction:**
   Key melodic elements, including pitch, tempo, rhythm, and note durations, are extracted from MIDI files. These features are critical for modeling the melodic structure.
2. **MIDI File Parsing:**
   The pretty_midi library is employed to read MIDI files and extract structured musical data essential for training.
3. **Normalization and Sequence Padding:**
   Melodic features are normalized to a standard scale and padded to ensure consistent lengths for efficient processing by the model.

Train-Validation Split:
To evaluate the model's performance on unseen data, the dataset was split into training and validation sets. A split of 80% for training and 20% for validation was employed, striking a balance between robust model training and sufficient evaluation data.

Models Approach
both approaches utilizes a Long Short-Term Memory (LSTM) network. LSTMs, a type of Recurrent Neural Network (RNN), excel at capturing temporal dependencies in sequential data, making them particularly suitable for this task of combining lyrical and musical features to generate predictions. Below is a detailed breakdown of each approach and the corresponding architectural structure.

Approach 1: Integrating General MIDI Features

The first approach integrates general MIDI features that capture structural and timing-related information from the melodies. These features include:

- **Resolution:** Represents the number of ticks per time frame.
- **Key Signature Changes:** Tracks shifts in the musical key.
- **Timing Features:** Includes tick_to_time (mapping ticks to actual time) and tick_scales (scaling ticks to maintain consistent temporal representation).

Data Preprocessing:

- MIDI features are processed to ensure consistency across samples. This includes trimming or padding sequences to a fixed size of 100 dimensions.
- The processed MIDI features are combined with the pre-trained Word2Vec embeddings of the lyrics using np.column_stack. This creates a unified input format, where the structural MIDI data is aligned with the corresponding lyrics.

Approach 2: Integrating Sonic MIDI Features

The second approach enhances the model by incorporating sonic MIDI features, which describe the sound characteristics of the melody. These features include:

- **Pitches:** Frequencies of the notes.
- **Velocities:** Intensities of the notes.
- **Durations:** Length of each note.
- **Instruments:** Types of instruments used in the melody.

Data Preprocessing:

- Sonic features are processed and scaled to maintain uniformity across samples. Similar to the first approach, sequences are padded or trimmed to a fixed size of 100 dimensions.
- The processed MIDI features are combined with the Word2Vec embeddings of the lyrics using np.column_stack, enabling the model to simultaneously learn lyrical and sonic patterns.

Model Architecture: LSTM with Multihead Attention

The proposed architecture integrates an LSTM network with a Multihead Attention mechanism to effectively predict song lyrics by capturing relationships between lyrics and melodies. The model combines Word2Vec embeddings for lyrics with MIDI features to create a comprehensive input representation. Below is a detailed breakdown of the architecture:

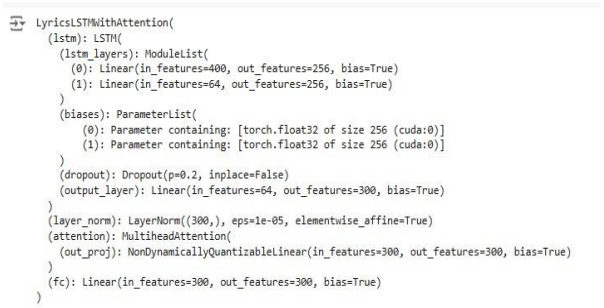
Architecture Components:

1. **LSTM Layers:**
    - Two LSTM layers process sequential input data to capture temporal dependencies. ○ The first LSTM layer accepts a combined input size of 400 features
      (Word2Vec embeddings and MIDI features) and outputs 256 features.
    - The second LSTM layer processes a smaller input of 64 features and outputs 256 features.
    - Bias parameters are included in each LSTM layer to improve learning capabilities.
2. **Dropout Layer:**
    - A dropout layer with a rate of 0.2 is applied to prevent overfitting by randomly deactivating a fraction of the network connections during training.
3. **Multihead Attention:**

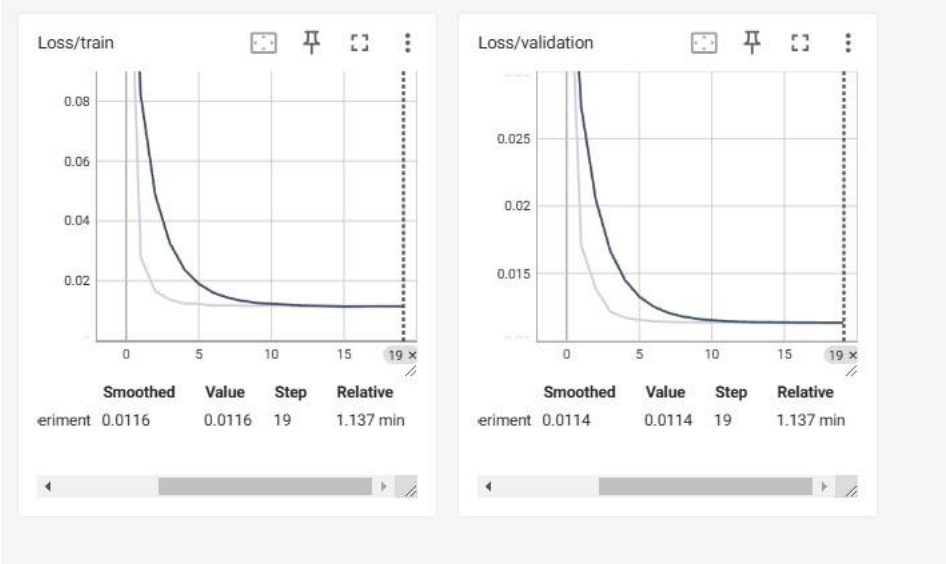
- The attention mechanism includes four heads, enabling the model to focus on different parts of the sequence simultaneously. ○ This component improves the model's ability to understand contextual relationships between lyrics and melody.

4. **Layer Normalization:**
   - A normalization layer with 300 features ensures stable training and prevents gradient explosion.
5. **Fully Connected Layer:**
   - The final dense layer maps the processed data to the output space. ○ This layer is responsible for predicting the next word in the sequence, with input and output sizes of 300 features.
6. **Loss Function:**
   - Mean Squared Error (MSELoss) is used to evaluate the performance of the model by comparing predicted and actual word embeddings.
7. **Optimizer and Learning Rate Scheduler:**
   - The Adam optimizer with a weight decay of 1×10−51 \times 10^{-5}1×10−5 is employed for efficient training. ○ A ReduceLROnPlateau scheduler dynamically adjusts the learning rate based on validation performance.
8. **Gradient Clipping:**
   - To stabilize training, gradients are clipped to a maximum value of 1.0.

The alignment between lyrics and melody is facilitated by the collate_fn function, which processes batches of data. This function consolidates the inputs and targets from each item in the dataset and returns them as tensors.
For example, inputs represents the combined input (lyrics and melody), while targets contains the prediction targets, which are the word vectors for the next timestep. This function ensures that the data structure is fully compatible with the model's requirements during training.

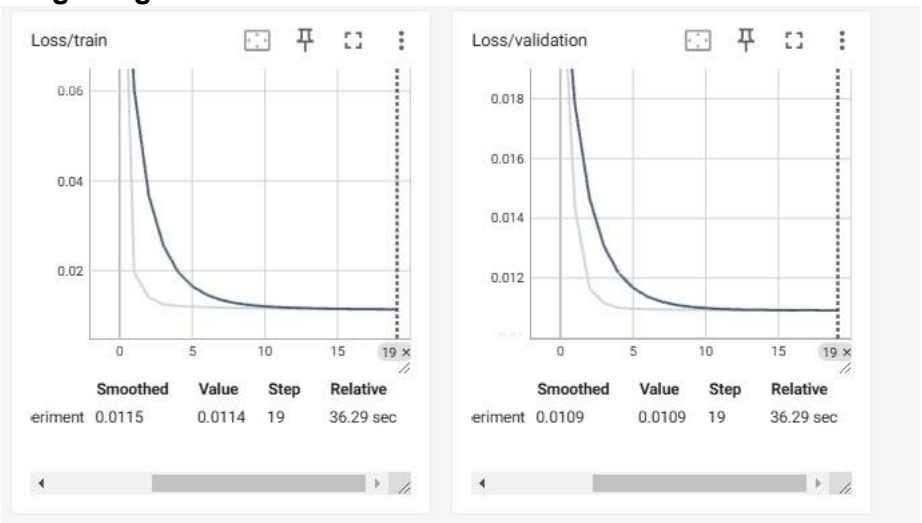


C.
Integrating General MIDI Features



.

Integrating Sonic MIDI Features



Graph analysis

The train graph shows us a decrease in loss occurring between 0.08 and 0.01, which indicates the model's improving skill. Learning occurs over time. The downward trend reflects the model's improved ability to understand and predict patterns within the external data set. The steady decline signifies that the model is successfully learning from training data, improving its prediction accuracy as it is given more information. The improvement in the val sample shows a decrease in LOSS, indicating that the model is learning efficiently from the training data and successfully applying its learned insights to words and melodies.

7.

I decided to build a class called **vocabulary**, which was designed to manage the relationship between the string representation of a word and its corresponding embedding vector. My focus was on capturing the contextual nuances of the model's attentional features. During the word generation process, the model used embeddings based on a sequence of words. At each step, the next word was selected based on its similarity to the previous embeddings.

To prevent repeated outputs, I implemented a penalty mechanism that reduced the score for words that were used recently in the sequence. This mechanism also prevented excessive reuse of words that had already appeared more than ten times. In addition, the use of a temperature parameter allowed for the regulation of randomness in word selection, while balancing diversity and coherence.

To evaluate the model's performance, I used four different metrics:

1. Textual similarity calculation:

- Cosine similarity

- Word mean distance (WMD)

- Sentence-BERT (SBERT)

2. Lexical overlap:

- Measuring the percentage of words in common between the original text and the generated text.

3. Repetition ratio:

- Calculating the proportion of unique words relative to the total word count in the generated text.

4. Quality Score Calculation:

- The final quality score is derived using a weighted formula, distributing importance across metrics as follows: - Cosine Similarity: 30%

- Lexical Overlap: 20%

- Word Moving Distance (WMD): 30%

- Semantic Similarity (SBERT): 20%

- Repetition Ratio: 10%

This systematic approach ensured a robust assessment of a text from more than one viewpoint a.

Integrating General MIDI Features:

osine similarity: 0.9174 Vocabulary overlap: 0.0575 Word Mover's Distance: 2214.0463 Generated lyrics: forever guess going actually want something us know really know get ok going ok even like go guess suppose suppose suppose get think want us though even say actually though want want something guess ok guess like us like really think really say going really think say see something really us go even think actually like us really us really get us go us something even see actually actually see say something like actually think say really even say us think think even go go even even think see know us think say actually see really something something think see say

---

Melody: Melody 4: aqua - barbie girl, Version: 1, Temperature: 0.7

Cosine similarity: 0.9176

Vocabulary overlap: 0.0490 Word Mover's Distance: 2894.2860 Generated lyrics: forever something actually know think go see want go us ok really actually think us say like suppose even see guess suppose like know really going ok see say really going want though know get go really know see though going ok want get even say going get even get going something even ok say something even want say see ok really something even say going though really want guess though say ok something really ok see really want know even something see say something say ok really even ok guess really say ok though really though even like see want

---

Melody: Melody 5: blink 182 - all the small things, Version: 1, Temperature: 0.7

Cosine similarity: 0.9165

Vocabulary overlap: 0.0484 Word Mover's Distance: 1322.4393 Generated lyrics: forever actually want see guess see suppose going actually ok going go even say say really get know like something suppose ok us think even go though like really get ok us guess suppose ok think know want even ok suppose us really guess go guess us get even think something get really like want see really get like really even think want like guess ok go want think something get going even actually go like like even something really really though go though get something get really want want want like even know even know something think guess think go

---

Results for initial word: rhythm

Melody: Melody 1: the bangles - eternal flame, Version: 1, Temperature: 0.7

Cosine similarity: 0.9163

Vocabulary overlap: 0.0339 Word Mover's Distance: 715.6271 Generated lyrics: rhythm get say see even go us ok something us ok guess get know though going like actually think want see something something us guess suppose see think actually even suppose going really get want say though going even even though want guess say want going though know go suppose say go know like guess go really say really want say want going really guess say though really want know ok though know guess go going go want guess really say go though want think though go like though like go guess say really want say really really say want know

---

Melody: Melody 2: billy joel - honesty, Version: 1, Temperature: 0.7

Cosine similarity: 0.9169

Vocabulary overlap: 0.0303 Word Mover's Distance: 1595.7672 Generated lyrics: rhythm ok even really go think get actually want suppose say go know going know guess ok though though us something like actually see guess think want ok want think really say even even like get see like though think see us us guess get really get something suppose ok something really actually going something ok going really say something suppose actually something going suppose actually really ok see really get going actually actually going going going actually want actually say know go say get going suppose really actually going suppose something ok actually want get get suppose suppose even get

---

Melody: Melody 3: cardigans - lovefool, Version: 1, Temperature: 0.7

Cosine similarity: 0.9165

Vocabulary overlap: 0.0575 Word Mover's Distance: 2216.2490 Generated lyrics: rhythm say going want actually think get though guess us something even suppose like go even actually really say really say get see us going guess going though think us actually want go like get actually go going something guess suppose get think us see guess ok see suppose want though suppose guess think suppose something actually ok something want though though think suppose guess ok like guess know say get think know know want want know think want ok think know go though guess like get guess get want get suppose think suppose get think like guess though want think

---
Melody: Melody 4: aqua - barbie girl, Version: 1, Temperature: 0.7
Cosine similarity: 0.9169

Integrating Sonic MIDI Feature:

Melody 1: the bangles - eternal flame Original
lyrics:
close your eyes give me your hand darling & do you feel my heart beating & do you understand & do you feel the same & am i only dreaming & is this burning an eternal flame & i believe it's meant to be darling & i watch you when you are sleeping & you belong with me & do you feel the same & am i only dreaming & or is this burning an eternal flame & say my name sun shines through the rain & a whole life so lonely & and then you come and ease the pain & i don't want to lose this feeling &

Similarity scores:

Initial word: heart
Version: 1
Temperature: 0.7
Cosine similarity: 0.9166
Vocabulary overlap: 0.0339
Common words: heart, want
Word Mover's Distance: 715.4264
Semantic similarity: 0.1680
Repetition ratio: 0.1980 Weighted score: 0.3356 Generated lyrics: heart suppose think get going know suppose ok even going want something like know really could though ok thats even ya think something guess could want actually though ok thats even think get like something go even though suppose actually really guess go suppose guess ok want get get going even really go like though see want could could something guess like like ya ok though ok ok actually see think actually ok ok though want see think though though ya really want ya ok ya actually actually think thats ya ok though thats thats see though ya actually think something
---
Initial word: freedom
Version: 1
Temperature: 0.7
Cosine similarity: 0.9165
Vocabulary overlap: 0.0169
Common words: want
Word Mover's Distance: 713.5737

Semantic similarity: 0.0885
Repetition ratio: 0.1980 Weighted score: 0.3163 Generated lyrics: freedom something guess something ok ya really think want like like even think even see going ok go know thats see ya like get get could know even go going could going actually suppose suppose ya could actually know ya could could actually really know really really guess though go guess something guess go actually though want want ya thats ok actually though though ya though really know really guess actually go go actually guess go want guess really ya go know go thats guess actually go want know actually actually know know know really though ya could ya actually go
---

Initial word: light
Version: 1
Temperature: 0.7
Cosine similarity: 0.9162
Vocabulary overlap: 0.0169
Common words: want
Word Mover's Distance: 716.5130
Semantic similarity: 0.1566
Repetition ratio: 0.1980 Weighted score: 0.3298 Generated lyrics: light know really really could guess like want even thats though going actually thats ya see guess want like something though think go actually get even ok suppose going go though something know like ok get could though get see something actually ok know though even ya even ya going know think like get could see thats could though know though ok actually like see think see ok think even could suppose ok like guess actually think really actually see suppose guess like actually see think see suppose guess think like suppose see suppose actually something think suppose ok going something
---

Initial word: forever
Version: 1
Temperature: 0.7
Cosine similarity: 0.9156
Vocabulary overlap: 0.0169
Common words: want
Word Mover's Distance: 712.4827
Semantic similarity: 0.1932
Repetition ratio: 0.1980
Weighted score: 0.3369 Generated lyrics: forever like go ya want get something even actually actually know actually think even thats going guess get think going go really want get see really something see though ok like suppose suppose suppose suppose ya guess something like like want thats know guess go could thats guess know ya know something ya know though even go want ya go guess though ya go guess could ya though ok guess ok something think

though know though want though know guess go ya going go guess something guess go something go guess go could ya though ok thats ok know ok though

---

Initial word: rhythm

Version: 1

Temperature: 0.7

Cosine similarity: 0.9168

Vocabulary overlap: 0.0169

Common words: want

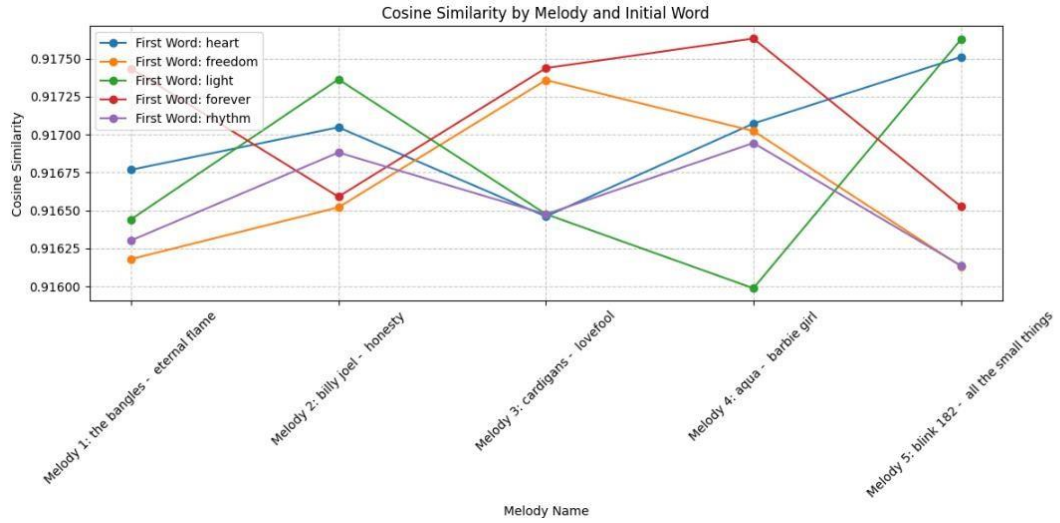
Word Mover's Distance: 711.8264

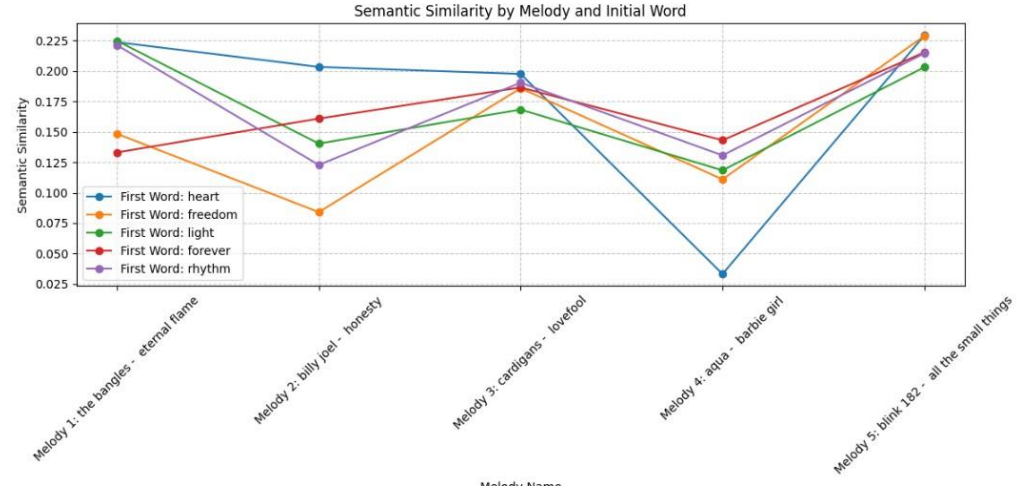
Semantic similarity: 0.1668

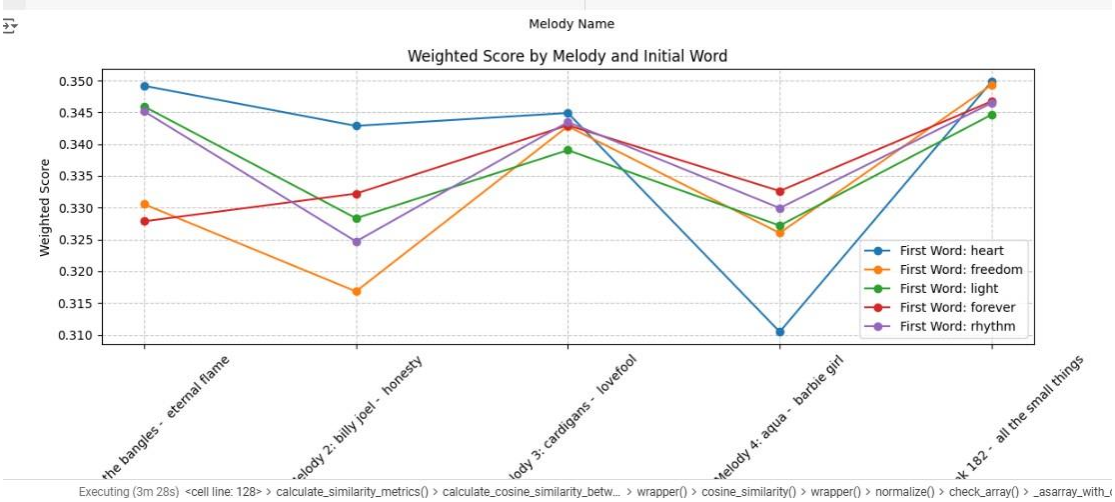
Repetition ratio: 0.1980 Weighted score: 0.3320 Generated lyrics: rhythm actually could think ok something think actually even really suppose really ok get get really want something ok thats ya guess want going like ya thats suppose though going see guess suppose could ya see thats go even know ya could go though go guess could thats like see suppose even know guess think like going actually suppose going guess thats could thats even see going guess even suppose though know though even go could though suppose ya even like thats like see thats see though though even even could know ya even ya thats like see even know like

---

c.







The initial word significantly impacts the generated lyrics by establishing the thematic and semantic direction of the sequence. LSTM models rely on sequential

dependencies, meaning the initial input heavily influences subsequent predictions.

Differences arise because certain words, like *"heart"*, align more closely with common themes in song lyrics, such as emotions and relationships, while abstract terms like *"freedom"* or *"light"* may lead to more diverse and less thematically consistent outputs. This distinction underscores the importance of selecting an initial word that resonates with the emotional and thematic context of the melody to enhance the coherence and quality of the generated lyrics.