**Report – Assignment 3**

**Yael Berkovich - 324879501**

**Yuval Tzur - 206542839**

**Dataset Analysis & Processing:**

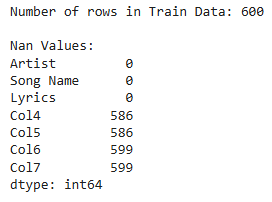
**Train Dataset** – / content / lyrics\_train\_set.csv  
Contains 600 rows and 7 columns:

**Structure of Raw Train Dataset and some examples:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **col 1** | **col 2** | **col 3** | **col 4** | **col 5** | **col 6** | **col 7** |
| elton john | candle in the wind | goodbye norma jean & though i never knew you at all & …… | NaN | NaN | NaN | NaN |
| cat stevens | moonshadow | [chorus:] & oh i'm bein' followed by a moonshadow moon shadow moonshadow--- & leapin … | morning has broken | morning has broken like the first morning & blackbird has spoken like … | NaN | NaN |
| billy joel | movin' out | anthony works in the grocery store & savin his pennies for some day & mama leone left a note on the door & she said "sonny move out to the country" … | it's still rock and roll to me | what's the matter with the clothes i'm wearing? & can't you tell that your tie's too … | big shot | well you went uptown riding in your limousine & with your fine park avenue &… |

We can observe 7 columns, where it is clearly visible that:

* Column 1 represents the **artist's name**.
* Column 2 represents the **song name**.
* Column 3 contains the **song's lyrics**.
* Column 4 mostly contains missing values, but occasionally includes an **additional song name**.
* Column 5 also has many empty values, but in some cases, it contains the **lyrics of the additional song**.
* Column 6 contains a non-missing value in only one row across the entire dataset, which is an **additional song name**.
* Column 7 contains the **lyrics of the additional song**.

NaN Values:

**Fixing the data:**

Rows that contain non-NaN values in columns 4, 5, 6, and 7:  
We split the row into separate rows:

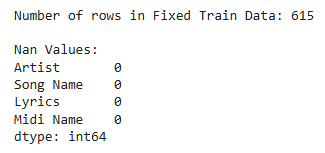
1. The **first row** will retain the original values in columns 1, 2, and 3 (Artist Name, Song Name, and Lyrics).
2. The **second row** will include values from columns 1, 4, and 5 (Artist Name, Additional Song Name, and it's Lyrics).

If there are values in columns 6 and 7:

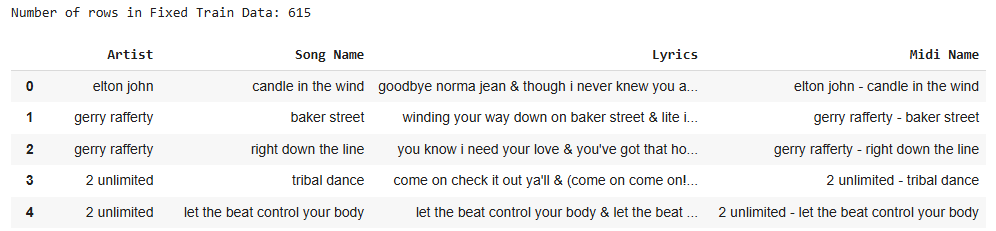
1. The **third row** will include values from columns 1, 6, and 7 (Artist Name, Additional Song Name, and Lyrics).

After this correction, the Train Data contains **615 rows**, and **7 columns**.  
We will name the first 3 columns : [**'Artist'** , **'Song Name**' , **'Lyrics'**]   
Columns 4-7 have now NaN / None only, so we can drop them.

To match the records in the dataset with the files in the midi\_files folder, we will create a new column named '**Midi Name**', which will contain the artist's name and the song title in format: {Artist} – {Song Name}

NaN Values after fixing:

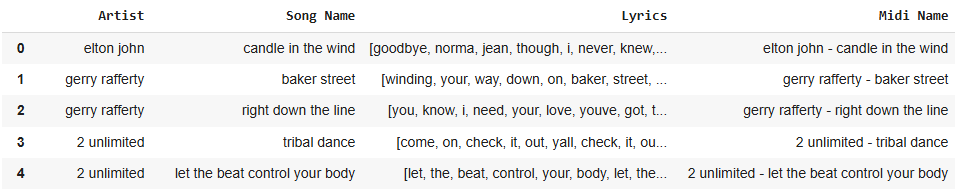
**Train Data after fixing:**



**Lyrics Cleaning:**

Now, we will clean all the words in the 'Lyrics' column by removing unwanted characters, non-word symbols, duplicate spaces, etc...

After cleaning, we will update the 'Lyrics' column to be a **list of clean tokens**.

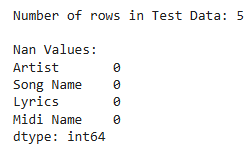
**Train Data after cleaning:**

**\**

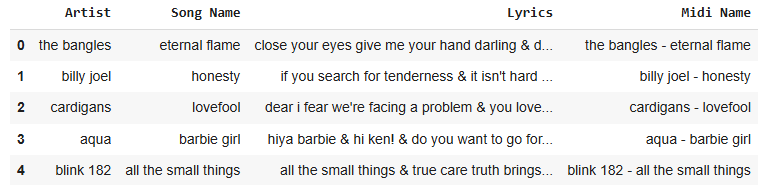
**Test Dataset** – / content / lyrics\_test\_set.csv  
Contains 5 rows and 3 columns:

**Structure of Raw Test Dataset, and some examples:**

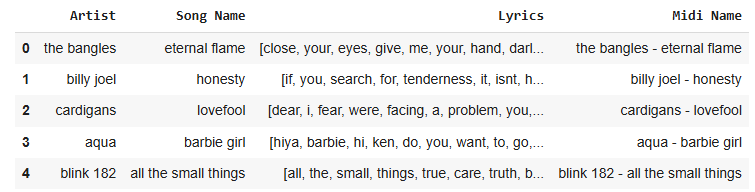
|  |  |  |
| --- | --- | --- |
| **col 1** | **col 2** | **col 3** |
| the bangles | eternal flame | close your eyes give me your hand darling & do you feel my heart beating & do you understand & do you feel the same & am i only dreaming … |

NaN Values:

We will name the columns: [**'Artist'** , '**Song Name**' , '**Lyrics**'] , and then we will create a new column named '**Midi Name**', which will contain the artist's name and the song title in the format: {Artist}– {Song Name}

**Test Dataset:**

Now we will **clean** and **tokenize** the lyrics as described previously in Train Data.

**Test Data after processing:**

**Download Pretrained Model:**Download and save the model at "/content/word2vec-google-news-300.model"

****  
The goal of this model is to leverage a pre-trained word embeddings dictionary to enhance the representation of words and improve the model's performance.

**Create Embeddings:**

1. Extract Unique Words from the Lyrics

The lyrics from the train\_data and test\_data DataFrames are concatenated into a single collection, a list is created with all words from these lyrics and the unique words are then identified and sorted.

The result is a list of unique words (unique\_words) that will later be used.

1. Generate Word Embeddings

The function create\_embeddings checks if each unique word is present in the pre-trained model (pretrained\_model) we have downloaded earlier.  
If a word exists in pretrained\_model, its corresponding embedding is retrieved and stored in the word\_embeddings dictionary.

The result is a dictionary word\_embeddings containing words as keys and their corresponding embeddings as values.

1. Filter Tokens to Match the Pre-trained Embeddings

The filter\_tokens function filters out words in the 'Lyrics' column that are not present in word\_embeddings.  
This ensures that only words with valid embeddings remain in the lyrics data.

The result is cleaned lyrics in both train\_data and test\_data, where all words have corresponding embeddings.

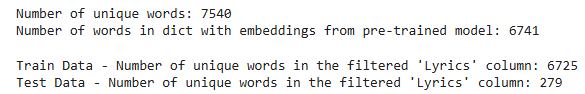
1. Create Embeddings for Each Song's Lyrics

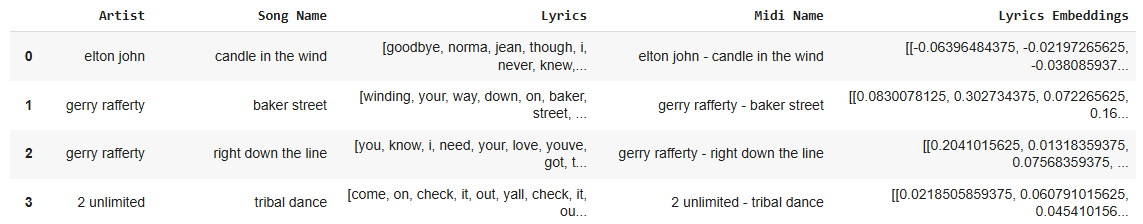
The function create\_lyrics\_embeddings iterates over each song in the dataset using its Midi Name as a key.  
For each word in the lyrics, the corresponding embedding from word\_embeddings is appended to a list.  
The list of embeddings for the lyrics is stored in a new column, '**Lyrics Embeddings'**.

The result is that each song now has its lyrics represented as a sequence of word embeddings, and these embeddings will be used later as inputs to the model for training and testing.

Statistics:

* **7540 unique words**, but only **6741 of them has corresponding embeddings** from the pretrained model.
* After filttering train and est datasets, there are:
  + **Number of unique words in train\_data [ 'Lyrics' ] : 6725**
  + **Number of unique words in test\_data [ 'Lyrics' ] : 279**



**Updated Train Data:**

**Updated Test Data:**

Now, in the 'Lyrics Embeddings' column, each entry is a nested list where the outer list corresponds to the number of tokens in the 'Lyrics' column, and each inner list contains 300 elements representing the embedding of each token.

**Processing Data and MIDI Files:**

1. Extract zip into /content
2. Create a Set of Valid MIDI NamesThe Midi Name columns from both train\_data and test\_data are combined into a single set, valid\_midi\_names, to ensure no duplicates.

This set represents the expected MIDI file names that should correspond to the files in the midi\_files directory.

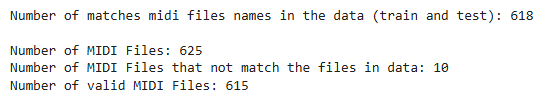
1. Load and Clean MIDI Filenames

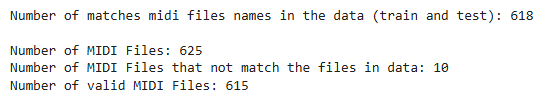
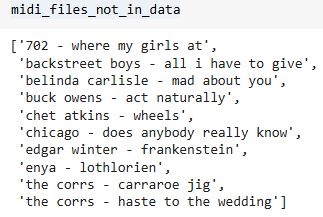
The filenames in the midi\_files directory are cleaned to align with the format of valid\_midi\_names.

Unwanted characters such as &, .mid, and extra spaces are removed, special characters are replaced, and all names are converted to lowercase for consistency.

Known discrepancies in three specific cases (e.g., "depeche mode - enjoy the silence-2") are manually corrected.

1. Compare Cleaned MIDI Filenames  
   The cleaned filenames are stored in a list, cleaned\_midi\_files.  
     
   Files that do not match any name in valid\_midi\_names are identified as midi\_files\_not\_in\_data.  
     
   Files that successfully match are stored as valid\_midi\_files.

 **Statistics:**

Non-Matching MIDI Files:

**Extracting & Processing MIDI Features:**

1. Add Melody Features

We calculate the **average word duration** to estimate the timing of when each word is sung in a song.   
Since the lyrics do not come with explicit start and end times, and the melody features (e.g., pitch and velocity) are indexed by timing, we need a way to **align these features** with the words in the song.

By dividing the total duration of the song (from the MIDI file) by the number of words in the lyrics, we estimate the duration of each word.   
This allows us to associate each word with a time window, enabling the extraction of pitch and velocity features that occur within that window.

**The goal** of this process is to incorporate the melody's musical context into the word embeddings, enhancing the model's understanding of the song.

* **Approach 1 – Pitch & Velocity features**
* **Pitch** provides insights into the melody's tonal characteristics, which can guide the model in selecting words that match the emotional tone of the music (e.g., deeper tones for somber words, higher tones for lively words).
* **Velocity** reflects the intensity and energy of the notes, helping the model align the pacing and dynamism of the lyrics with the melody.

By integrating these features, the model can better capture the interplay between lyrics and melody, resulting in lyrics that are more contextually aligned with the musical structure of the song.

**Approach 2 – Adding Instruments features**

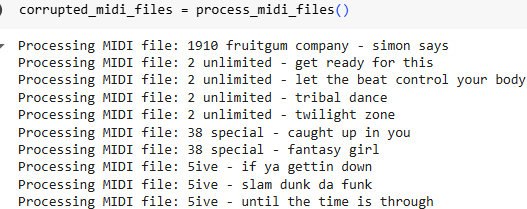
In addition to Pitch & Velocity, incorporating **Instrument features** provides a more comprehensive understanding of the musical context.   
The instrument features represent the dominant instruments used in the melody and their characteristics, which can significantly enhance the alignment of lyrics with the music's mood and genre.

**Instrument Context:** Each instrument has its own tonal quality (e.g., strings convey warmth and emotion, while percussions add rhythm and intensity).   
By identifying and incorporating instrument information, the model can generate lyrics that resonate with the overall orchestration.

**Dynamic Adaptation:** Instruments often drive the transitions in musical sections (e.g., the introduction of a violin in a soft passage or drums in an energetic chorus).   
Capturing this dynamic enables the model to craft lyrics that adapt to these transitions seamlessly.

**Genre Influence:** Instruments can hint at the musical genre (e.g., electric guitars for rock, synthesizers for electronic music), enabling the model to align the lyrics stylistically with the melody.

This multi-feature approach provides a richer dataset and enables the model to generate lyrics that feel more naturally integrated into the melody, catering to diverse musical genres and emotional contexts.

1. Process MIDI Files  
   **Valid MIDI files** are loaded using 'pretty\_midi', and corrupted or unsupported files are logged.  
   For each valid MIDI file, pitch and velocity features are calculated using the 'align\_features\_and\_lyrics' function.   
   **In Approach 2, Instruments features are calculated in addition to pitch ang velocity features.**  
    **The features are stored in dictionaries** based on whether the file corresponds to the train or test dataset.
2. Remove corrupted or unsupported files
3. Normalize Features  
   The features are **flattened** into a single list and **normalized** using Min-Max scaling to the range [0, 1].   
   The normalized values are split back into dictionaries for features, ensuring consistency across the dataset.
4. Sum Word Embeddings with Features  
   Normalized features are expanded to match the dimensionality of the word embeddings (300 dimensions) and added to the embeddings.   
    **Approach 1** :  
   Embeddings = lyrics embeddings + normalized pitch + normalized velocity

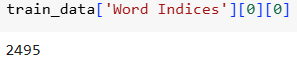
**Approach 2 :**  
Embeddings = lyrics embeddings + normalized pitch + normalized velocity + normalized instrument

The enriched embeddings are stored in the '**Embeddings**' column of the train and test datasets.

1. Create One-Hot Vectors  
   Unique words in the dataset are mapped to numerical indices (word\_to\_index), which are then used to generate one-hot encoded vectors.   
   The word indices are stored in the '**Word Indices'** column, and target sequences are stored in the '**Labels'** column.

Now, the '**Word Indices**' column contains a list of indices for each word in Lyrics.  
For example:  
The first word in 'Lyrics', in the first row of train\_data - is '**goodbye**'. This word is stored in the **2495** place in   
the dictionary of unique words (word\_embeddings):

תמונה שמכילה טקסט, גופן, צילום מסך, קו

התיאור נוצר באופן אוטומטי

1. Generate Sequences and Targets  
   Sliding windows of fixed size (**window\_size**) are applied to generate overlapping sequences of word embeddings or indices.   
   Corresponding target sequences for the **next words** in the sequence are created and stored in columns:  
   '**Window Word Embeddings'** , **'Word Index**' , **'Next Word Index**'
2. Create One-Hot Encoded Labels

* **Due to limited RAM, we saved the datasets without 'Labels' column in Parquet and Pickle formats. After loading them, we add the 'Labels' column.**

For each **next word index**, one-hot encoded vectors are created using the total number of unique words. The resulting one-hot encoded labels are stored in the '**Labels**' column of the datasets.

For the model, we will use 2 columns for training:  
- 'Window Word Embeddings'  
- 'Labels'

For the generating of the lyrics, we will use additionally the columns:  
- 'Pitch Features'  
- 'Velocity Features'

**The model:**

We decided to use the LSTM network combined with a layer of soft max and dropout.

**Recurrent Layer (LSTM)**

The core of the model is an LSTM network.   
The LSTM is responsible for capturing sequential dependencies in the input data, both in the text and the melodic features. The configuration of this layer includes:

* Input Size: 300 (embedding dimension) with added dimensions for pitch and velocity.
* Hidden Size: 128 units, which determines the size of the hidden state vector.
* Dropout: 20% , applied to prevent overfitting.

The LSTM processes the combined input sequentially, maintaining a hidden state that summarizes the context of the sequence up to the current step. This enables the model to generate contextually relevant predictions.

**Fully Connected Layer**

The output from the LSTM is passed on to a fully connected layer (nn.Linear).   
This layer converts the 128-dimensional hidden state into a vector of size equal to the vocabulary.   
The model maps the learned contextual representation of the input sequence to logits corresponding to each word in the vocabulary. It basically provides a linear projection that the subsequent softmax layer can convert into probabilities for word prediction.

**SoftMax Addition**

The final process of the model is a softmax calculation and it's called in the generating lyrics function.   
The layer converts the logits from the fully connected layer into a probability distribution.   
This distribution enables a probabilistic interpretation of the model's outputs, where each value represents the likelihood of a specific word being the next in the sequence.   
Additionally, temperature adjustments can be applied during sampling to control the diversity of the generated text.   
Lower temperatures make predictions more deterministic, focusing on the highest-probability words, while higher temperatures encourage diversity by increasing the likelihood of less probable words.   
This distribution indicates the likelihood of each word in the vocabulary being the next word.   
During generation, the model samples from this distribution to determine the next word in the sequence.

Loss Function -Cross-Entropy LossOptimization –

* Optimizer: Adam optimizer is used with a learning rate of 0.001.
* Gradient Clipping: Gradients are clipped to prevent exploding gradient issues.

Regularization -

* Dropout: Dropout is applied to the LSTM output to prevent overfitting.
* Early Stopping: Training stops if the validation loss does not improve for a specified number of epochs.

**Custom Collate Function: custom\_collate()**The custom\_collate() function is a function designed to prepare batches of variable-length sequences for training and evaluation. It ensures that all sequences in a batch are **padded** to the same length, enabling efficient processing by the LSTM model.

* **Input:**A batch of data, where each item is a tuple containing a **sequence** of input data and corresponding **labels**.
* **Process:**The function separates the input data and labels from the batch into two separate lists.  
  The input data and labels are padded using the pad\_sequence() function from PyTorch, ensuring that all sequences in the batch have the same length, with shorter sequences being padded with zeros.  
  The padded input data and labels are returned as a batch for processing by the model.
* **Output:**A batch of padded data and labels, ready for input into the model.

**Training Process: train\_model()**The train\_model function is designed to train the lyrics generation model effectively by handling both training and validation phases, incorporating techniques to optimize and regularize the training process.

**Input Parameters:**

* Model: The LSTM-based lyrics generation model.
* Train Loader: DataLoader for the training dataset.
* Validation Loader: DataLoader for the validation dataset.
* Criterion: The loss function (Cross-Entropy Loss).
* Optimizer: The optimization algorithm (Adam optimizer).
* Num Epochs: The number of epochs for training.
* Early Stopping Patience: Number of epochs to wait for improvement in validation loss before stopping training early (default: 5).

**Process:**

First, we initialize a SummaryWriter that is used to log training and validation losses for visualization in TensorBoard, as well as gradient scaling (GradScaler) for mixed-precision training to improve computational efficiency.  
  
**Training phase:**The model is set to training mode using model.train()  
For each batch of training data:

* Sequence lengths are calculated to handle variable-length inputs.
* Data and labels are moved to the appropriate device (CPU/GPU).
* Predictions are generated by passing the input through the model.
* The loss is computed, and invalid indices (e.g., padding tokens) are excluded from loss computation.
* Gradients are calculated using loss.backward(), scaled, and clipped to prevent exploding gradients.
* The optimizer updates the model parameters.

The average training loss for the epoch is logged.

**Validation phase:**The model is set to evaluation mode using model.eval(), No gradients are computed (torch.no\_grad()), and the loss is calculated for the validation dataset.   
Afterward, the average validation loss is logged for the epoch.

We then checkif the validation loss improves, the best validation loss is updated, and patience is reset. However, If the validation loss does not improve for early\_stopping\_patience epochs, training stops early to prevent overfitting. Training and validation losses are logged to TensorBoard for visualization, and the time elapsed during training is printed for tracking progress.

**Output :**  
The trained model with logged metrics for analysis.

**Lyrics Generation Process – generate\_lyrics()**

The generate\_lyrics() function is responsible for generating lyrics based on an initial seed word and melodic features ('Pitch Features', 'Velocity Features').

**Input Parameters:**

* model: Our trained LSTM model for lyric generation.
* start\_word (str): Starting word for generating lyrics.
* word2embedding (dict): Mapping from words to their embeddings.
* index2word (dict): Mapping from indices to words.
* data (dict): Dictionary containing the current row's data, such as "Pitch Features" and "Velocity Features".
* max\_length (int): Maximum number of words to generate.
* temperature (float): Sampling temperature for diversity control.

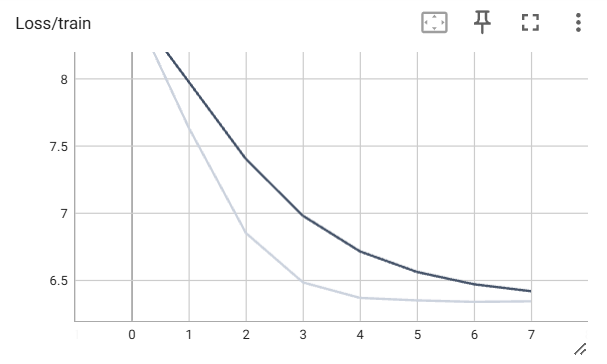
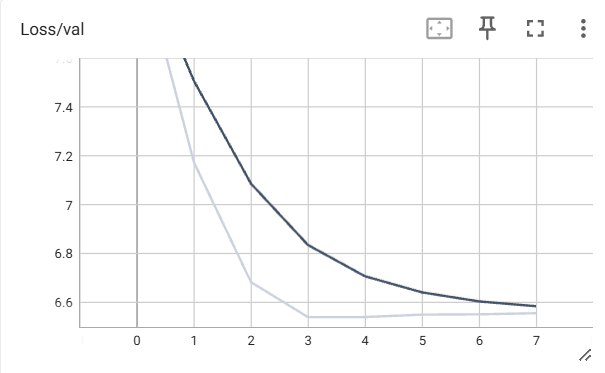
**Process:**The function begins by initializing the embedding of the initial word and integrating it with the corresponding melodic features (pitch and velocity).  
A recurrent loop predicts the next word iteratively by feeding the current input (word embedding combined with melodic features) into the LSTM.   
The LSTM generates a hidden state, which is passed through a fully connected layer to produce logits for each word in the vocabulary.

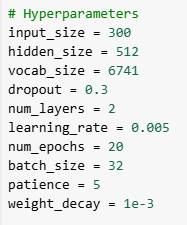
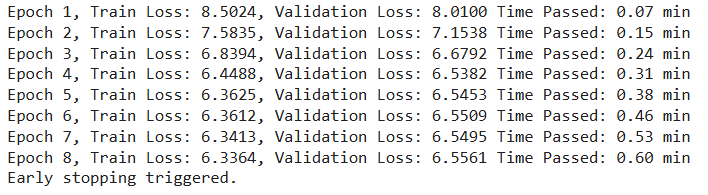
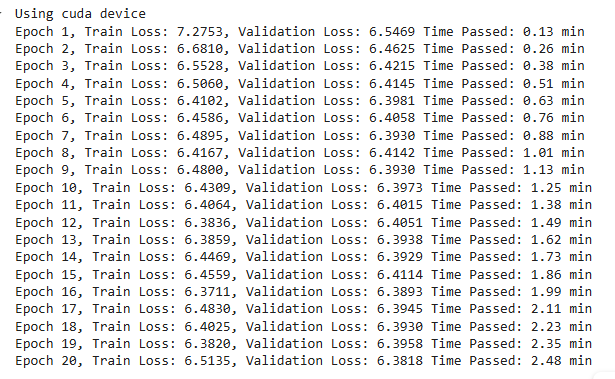
The softmax layer converts the logits into a probability distribution, and the next word is sampled based on these probabilities.   
Pitch and velocity values for the next timestamp are integrated with the sampled word's embedding to prepare the input for the next iteration.

This process continues until the sequence reaches the maximum length or an <END> token is generated.

**Output:**The function returns a string of generated lyrics, which are evaluated for coherence, fluency, and relevance to the melodic input.

**Results  
First Approach**

Tensorboard -

****  
The models training epoch's results:

**Lyrics Generation:**

**First word of the lyrics:**

Generated Lyrics for row 0 – the bangles – "eternal flame": **close**  
close waterloo bass dedicate anguish ugh hug inch madness antelopes verdict choice rustles taken miami dying pearls yelled carnival hash versatile snap taken plowshares al accidentally belongs childhood keeps active roller rockin higher trips ugh lazy bess envy repeated apple inside lazy

Generated Lyrics for row 1 - billy Joel – "honesty": **if**  
if electrical blastin dedicate completeness thats carnival miami his neat eloquence taken voodoo keepin astral hash measure meat ei miami mourn kiddin bellies keepin send friday yell table klar strikes ugh lazy quicker ugh his viene shines escorted treks bellies rustles appetite palms sporting athletic yelled ugh

Generated Lyrics for row 2 – cardigans – "lovefool": **dear**  
dear jenny beware cell twain taken lazy spoken gazed paris satin temperature hash dove fake buyin viene loss jet friday soot neat pops assure teach whispered mysterious yesterday spoken accidentally meet helping older locomotion miami underneath dove idol gifts feint profile quarter bellies whoa brooklyn miami compare

Generated Lyrics for row 3 – aqua – "barbie girl": **hiya**  
hiya steed based closest fostering probably guitar rite sporting taken eighteen silencio taken actor aim g ghost barren bellies palms erection g bedrooms repeated accidentally lazy beyond miami controlled boat backs redo wheels gifts appetite actor yea miami miami unopened ugh ugh neat person satin actor

Generated Lyrics for row 4 - blink 182 – "all the small things": **all**  
all satans strangeness miami yearning quit groups accidentally strikes anguish yelled yell lazy fantastic taken dare patch mighty search faking dove inside walkman miami mourn shoulder actor ugh ugh odds bettin higher neat earthquakes taken pane orgies ugh viene place someplace ugh weeks rustles clip palms ugh strikes pearls lawless

**Random Words:**

1. **" Love "**

Generated Lyrics for row 0 – the bangles – "eternal flame":   
**love** fantasies entangled appetite went lazy hospital drama now sleeved maintain viene sexual lazy appetite frightening miami shinning tearing completeness mourn pave lite soup lazy thats ugh anyways friday bellies havin sporting viene duck whipped ugh twistin nervous mysterious dip lieber bedroom plaques summoned treks insoles darken anguish jester thats

Generated Lyrics for row 1 - billy joel – "honesty":   
**love** tear irritates ugh temperature hash written thats yelled brite silencio ei redo ugh guessing mirage g treks belongs dove jail jail miami friday neat moe plaques written earlobe passin actor miami punching havin anguish tapped flying table slit darken miami plaques my miami reasons someplace frightening klar

Generated Lyrics for row 2 – cardigans – "lovefool":   
**love** nick grille are coonskin ugh playing person drama redo border farm know blowin send satans miami cribs candlestick tearing ers accidentally completeness ugh palms admire dangerous stayin crib hash cat spoken miami viene covered yelled dontcha caso ugh ugh lonelier swimmin cribs insanity outside ghost hash

Generated Lyrics for row 3 – aqua – "barbie girl":   
**love** gifts belongs unopened belongs santas samurai shorty escorted sloppy sabotage lawless mourn upper shines sleeve satin market went ugh belongs duck temperature closing sunlit wept philly reached humongous weeks coonskin kosher warmth force johnny kani hoodwink yelled twirl accidentally prove appetite anguish treating skis ground alegria leather

Generated Lyrics for row 4 - blink 182 – "all the small things":   
**love** tryin search punching upper stumblin ugh actor bum controlling ugh duck duck upper echoed distant actor anguish artis compare satin anguish anguish robbery covered coming reached prove underneath coffee thats groaning chameleons friday miami weed dove duck lazy ghost yelled redo billy fullest yelled barbie

1. **" drama "**

Generated Lyrics for row 0 – the bangles – "eternal flame":   
**drama** confronted legend love global gets fella send yea jester soot hair symphony landed anguish into whipped unh bellies san viene silencio reached kahlua stairs fitness miami miami speaks ugh choirs pops judge sane chi painful hash miami belongs redo sleepy wept miami spoken

Generated Lyrics for row 1 - billy joel – "honesty":   
**drama** eighteen fails amarillo avec duck yelled thats belongs angles ghost miami sent completeness duck proven whipped humongous bore sleepwalk chaperon off pops tapped plants zwei skis syne recipe clint case meat appetite friday miami attracting sammy nwa blistering alleyway affecting rustles imagining france needin noche ghost needin

Generated Lyrics for row 2 – cardigans – "lovefool":   
**drama** reached horses denies forgiven deftly palms miami bettin underneath showed playing brand arent pit plaques hash miami apple actor fingers viene soot nervous floating butterly hair keeps satin lots ugh dedicate covered rolls barricade arent working plants anguish seeing artis ugh sleeve balled hash powers ugh hitch japanese

Generated Lyrics for row 3 – aqua – "barbie girl":   
**drama** wept intelligence forced ease longest sloppy searching frontin al represent lurking dying accent whipped satans apple filling japanese bugging poke glorious unopened miami satin will are taken actor honeymoon ugh frown dove marx ghost bellies jailed bets types walkman ghost tuesday ei viene bathe tails signed havin ugh

Generated Lyrics for row 4 - blink 182 – "all the small things":   
**drama** aye fades rode meditation thanks outta portraits dove dukedom ugh starving troubled stairs represent appetite annoy whipped horns table sleeve drove shores earlobe suicide yelled triumphant yell childhood hash jimi robbery scream sammy la drei apple recreation tapped lazy miami table miami locomotion pantyhose ringin

1. **" control "**

Generated Lyrics for row 0 – the bangles – "eternal flame":   
control treks smacked capture silencio tight oz cue ghost old waking gazed apple keeps support run yell dedicate hash wave pinch camino maintain frightening appetite heavens bets noche voices place below whatever ihr went place anyways lazy spoken ugh hash kiddin sincere viene darken apple sloppy silencio letters

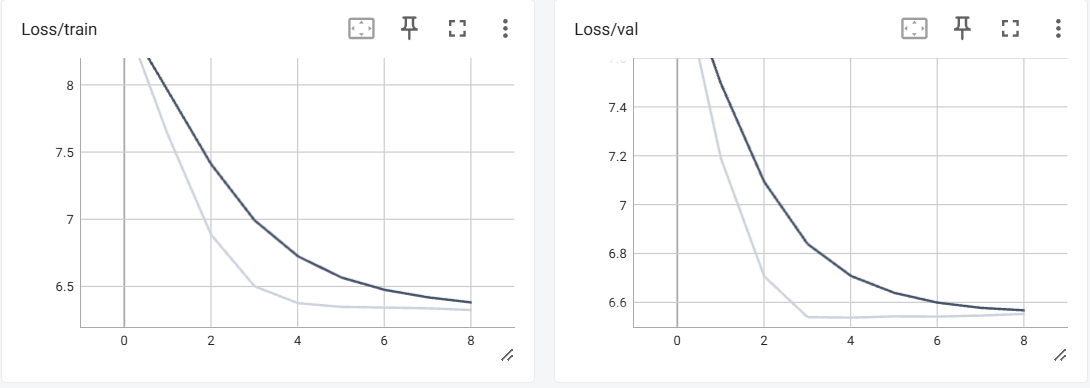
Generated Lyrics for row 1 - billy joel – "honesty":   
control purple spendin battle fitness miami completeness beyond pops hash written mamas ugh miami groaning bullied gifts ships spoken hateful ground palms backs lazy ghost spoken lite hair mental satin butterly baptized walkman yelled hairy lennon yesterday lazy unopened ugh profile measure dancing hazel hand

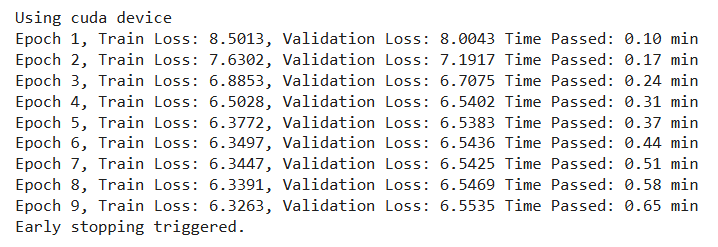
Generated Lyrics for row 2 – cardigans – "lovefool":   
control confused gegen flying wonderfully ei unopened not flamenco gifts dove shepherds before anguish endure date trail bettin vd earthquakes butterly keepin smacked sentenced property worlds slow ling groaning jail claim hash thats market anguish gully degenerate send taken soot nervous won went taken klar twistin

Generated Lyrics for row 3 – aqua – "barbie girl":   
control ugh jail mourn completeness waterloo actress g balled belongs ugh consent unopened eloquence stumblin prove dead reprobate carnival underneath spoken table pink claim dove earthquakes lieber represent miami ugh appetite shady tracks accidentally earthquakes packed yelled moonlit rustles pearls mistaken guile hash beware taken hazel ugh friday shepards sucks

Generated Lyrics for row 4 - blink 182 – "all the small things":   
control climbin slit childhood upper backyards bellies flying degenerate why groaning ugh higher miami drumstick actor babes palms bellies spire came freewheeling regret viene higher carrying soot ugh send miami soot lifeline san moral sleepy whipped sides starving prove offend lifeline spears dressed barbie pickin hash ugh bathe

**Second Approach**

Tensor board

  
The models training epoch's results:

**Lyrics Generation:**

**First word of the lyrics:**

Generated Lyrics for row 0 – the bangles – "eternal flame": **close**  
close outlaws twain snap colors gloves yell hash winding ihn inland viene bellies miami reign sleepwalk lazy actor murdered sloppy spoken insoles stairs blowin impale ugh yelled viene snap appetite repeated plight suffer techno thats bellies licks represent weeks chalet neat miami ghost shirt dove fevers treating repeated

Generated Lyrics for row 1 - billy joel – "honesty": **if**  
if indecisions taken coffee spoken smoked ugh grace chameleons sammy boogie miami groaning ruled miami floating dove taken insane madam prove suffer walkman rob windmill spite yell groovy ugh higher purple hurry yelled thugging cents endure viene keeps symbolic prime gully horns butterly strangeness done wildest sloppy

Generated Lyrics for row 2 – cardigans – "lovefool": **dear**  
dear accidentally loss pops neat america bellies viene place smacked pantyhose havin bellies miami rhyming yell caso apple drown miami force ei sporting choirs eloquence compares underneath ghost rumored g freaks butterly hash braille ruled dyin laughed butterly completeness lazy nwa claim bedrooms awhile sammy went

Generated Lyrics for row 3 – aqua – "barbie girl": **hiya**  
hiya retired hundred g refugees violent thugging consent lazy caso every wife kiddin nothings ugly tire miami kosher shepards zoot aire redo compares bedroom starving hash famous honeymoon sharing sammy appetite drama billy stairs unopened gifts viene lazy levee ei sloppy pave anguish miami hash

Generated Lyrics for row 4 - blink 182 – "all the small things": **all**  
all mysterious sony sloppy bedroom viene lazy dj ruff force sleeve backs accidentally g miami childhood cool shinning ugh inland treks spoken puss humongous yelled viene send monkey playing biggest anguish daughters actor ei glamor battle underneath hazel yelled battle plaques childhood hits fucks

**Random Words:**

1. **" Love "**

Generated Lyrics for row 0 – the bangles – "eternal flame":   
**love** underneath anguish underthe actor claim jedes yearning maintain lonelier appetite brand everyting emergency poke tearing completeness apple completeness prove unheard miami hash ghost satin lost japanese bedroom dedicate hurry year bronze viene lazy husbands jerking miami caso table viene holidays completeness heading possibilities shack miami search

Generated Lyrics for row 1 - billy joel – "honesty":   
**love** thats cock dove earthquakes actor strangeness anguish went coulda flying beginning spinning taken sloppy suicide friday bettin eloquence spark silencio ei stumblin ugh bellies rustles cockpit guile apple loss today went friday pops ugh miami missin granted clothing emergency now anguish ground neat lazy battle hide

Generated Lyrics for row 2 – cardigans – "lovefool":   
**love** lift jackson bop friday viene anguish dove gifts haste operation completeness actor dove ugh pops mellow climbin versatile observe naturally coaster hash symbolic insoles miami hair neat stairs viene hears honeymoon chalet accidentally walkman desires lazy round butterly vd pearls lazy miami hitch actor duck mistake

Generated Lyrics for row 3 – aqua – "barbie girl":   
**love** soul linen tommy lazy techno crowded endure hawk bellies bum psychiatrist miami taken miami wohl satin wok earthquakes friday anguish tapped taken butterly choirs dove ugh carnival remain built reputation chopped robbery bedroom lazy pops ugh miami ei tar mysterious boogie keepin tainted sloppy belongs swear viene soot

Generated Lyrics for row 4 - blink 182 – "all the small things":   
**love** stumblin lady braille ambition lazy klar shack picker father gifts table mourn force anguish soot prime device sporting pops size pops sales sexual ugh satin butterly nwa upper duck needin boogie hath powder slit g miami neat zoot auditorium satans mysterious dyin miami coupe lifeline sunlit

1. **" drama "**

Generated Lyrics for row 0 – the bangles – "eternal flame":   
**drama** learns expect camino bein pelicans plaques transmitted dyin written diamonds chrome jasmine heartache playing moment ruled whether thats tapped hoe anguish walkman upstairs slit miami scuff send duck appetite seventeen ugh crib support hoppin sony vd actor nwa actress duck inside weeks spoken guile thats sloppy wearin an

Generated Lyrics for row 1 - billy joel – "honesty":   
**drama** cada months vouch keep rides locomotion claim silencio sails yell winding extreme drown taken not ugh hash ever friars tear laid satan drown farmers spoken know lewis thats hurries provin stairs dedicate hitch chrome feed eminem sammy sits caso bettin ugh sleeve cents keep behave upper snap insoles

Generated Lyrics for row 2 – cardigans – "lovefool":   
**drama** alle energy hurl bunch trouper above gifts sleepwalk actor strange leon palms wonderfully billy sloppy fashioned bedrooms ships not playing mandy appetite horns blowin standin satin windmill bigger carrying truly wok granted despair anguish winding membership redo yells ugh temperature polluted thats ugh havin conceal ihr ghost

Generated Lyrics for row 3 – aqua – "barbie girl":   
**drama** kremlin vexes freudian beats invisible angles endure prove miami bellies ugh sharing zombie friday thats ugh strikes actor roar belongs groaning shines bullied gas spoken ugh windmill viene emergency temple taken roars redo passin appetite viene jokers threat dude spoken redo painful ugh thats bellies lazy

Generated Lyrics for row 4 - blink 182 – "all the small things":   
**drama** heel donner button asleep george havin claim shirt duck way yell echoes friday rash miami gained ugh springtime gauge monaco yelled table ugh ghost daughters powder button marvin blowin bettin viene silencio release tricks bettin innocence hash ugh drove bellies actor viene kiddin appetite prove his anguish

1. **" control "**

Generated Lyrics for row 0 – the bangles – "eternal flame":   
**control** gone pearls missin just accidentally ugh slit sure sleeve giddy accidentally his drown clinging class cheap honk boogie serious palms hash hit yelled billy appetite ugh viene yell dove japanese playing pearls ugh slate thats inside ugh donde tarry apple condom apple palms palms moonlight blowin deprived belongs

Generated Lyrics for row 1 - billy joel – "honesty":   
**control** hash jade pops redo patch went lazy portion actor satin ei sleeved ugh somethin sentenced walkman yell hair bellies slow convinced yelled tries dove viene ghost treks written tokyo mellow belongs blouses ugh bellies taken lyrical border inch soot dontcha silencio yelled

Generated Lyrics for row 2 – cardigans – "lovefool":   
**control** shines bellies between twain anguish yelled music helping hairy viene further high satin idol accidentally laundry appetite behave satin sat taken temperature carnival weird taken force miami miami stairs ugh written purple give competitive laid powder passin gifts mourn jean havin techno yesterday rustles boogie lazy drown sway hoppin

Generated Lyrics for row 3 – aqua – "barbie girl":   
**control** emergency almost ugh surrounding cry anguish palms nearest miami ugh completeness thats yell allen childhood moonlight wallet place drown brand taken morgan belongs carpets walkman actor ugh miami bathe repeated gifts ugh robbery thought ei redo bankroll monaco lazy know dove appetite hitch pantyhose

Generated Lyrics for row 4 - blink 182 – "all the small things":   
**control** miami cribs duke sleeved yelled taken tapped taken ugh deprived miami yelled lonelier most miami thats tock viene stairs silencio pops dove taken silencio anguish know havin thoughts firelight lonelier bellies palms miami pops accidentally hand measure viene not sleeve miami send tuesday inside sammy

**Insights on the Two Approaches**

**First Approach**

**Results**: The generated lyrics have a mix of meaningful sequences and random words.   
The use of pitch and velocity features adds variability but might not provide direct alignment with the melody.

* Random words for generation allow for diverse outputs, demonstrating model flexibility.

**Weaknesses**:

* The lyrics often lack coherence and semantic alignment with the melody.
* The generated sequences frequently include repetitive or nonsensical tokens.

**Second Approach**

**Results**: Similar to the first approach but with a slightly different strategy for incorporating features. This approach might emphasize the lyrics' alignment with the melody, given its structure.

* + Results show variation and some melody-context alignment.
  + Using embeddings for every token ensures consistent inputs.

**Weaknesses**:

* + The generated lyrics still have randomness and lack thematic consistency.
  + The reliance on token-level features can lead to overfitting to the embeddings rather than the melody context.

**Best Approach**:

* + The **Second Approach** appears more promising due to slightly better thematic alignment and coherence. While both approaches perform similarly in terms of randomness and diversity, the second approach's structured handling of embeddings might yield more interpretable outputs.

**Improvements:**

* + **Preprocessing**: Normalize or preprocess the pitch and velocity features to ensure they directly correlate with the model's understanding of context.
  + **Post-Processing**: Use a filtering mechanism to eliminate nonsensical phrases (e.g., using a language model to validate outputs).
  + **Feature Scaling**: Experiment with scaling the contribution of pitch and velocity to find the right balance between melody alignment and semantic richness.

**Future Work:**

* + Explore transformer-based architectures (e.g., GPT) for better long-term coherence in lyrics.
  + Fine-tune with more data on lyric-melody pairs to enhance melody-lyrics alignment.