Lyricade: An Integrated Acoustic Signal-Processing Transformer for Lyric Generation

Stanford CS224N Custom Project

Rhea Malhotra & Arnav Krishamoorthi & Klara Andra-Thomas
Department of Computer Science
Stanford University
rheamal@stanford.edu, arnavk@stanford.edu, klaraat@stanford.edu

Abstract

Lyric generation poses a multifaceted challenge in blending creativity, musicality, and linguistic skill to craft word schemes that resonate with the rhythm of a beat. Despite advances in generative models, the fusion of signal processing with transformer-based frameworks for fine-tuning has seen limited exploration, especially within acoustics. Here, we introduce Lyricade, an aid for lyric generation that leverages instrumental tracks and contextual cues (artist's name and song genre) through the fine-tuning of a Hugging Face GPT-2 model, underlined by signal processing via the Librosa library. Our methodology merges categorical song metadata, numerical audio features, and tokenized lyrics data for each song, facilitated through prompt engineering techniques. We evaluate three distinct model architectures, conducting extensive sweeps for feature weighting and hyperparameter optimization. Our front-end enables users to input any sound track and desired artist to iteratively generate lyrics for 10-15 second segments of a larger track, using the concluding lyric of each segment as a prompt for the subsequent one. Our final model culminates in a Training Loss of 0.75, BLEU score accuracy of 0.94, and a ROUGE-L score of 0.9752. See our demo here!

1 Key Information to include

- Mentor: Yuhui Zhang

2 Introduction

Lyric generation is as a complex problem within natural language processing, balancing creativity with the structured rhythm and flow inherent to music. Unlike traditional generative models, which focus on text-based outputs, lyric generation demands a sophisticated understanding of musical elements such as beat, rhythm, and the less tangible aspect of "flow." These requirements stretch beyond the capabilities of standard Generative Pre-trained Transformer (GPT) models, which lack the mechanism to process acoustic signals or understand musicality directly. Lyricade incorporates acoustic features alongside textual data for fine-tuning a lyric generation model.

The integration of signal processing and acoustic analysis into a transformer model framework presents significant challenges, primarily due to the inherent difficulty in merging numerical audio features with textual data in a way that enhances the model's generative capabilities. To this end, we leveraged the Spotify API and vast lyrics corpora to construct our own database.

The subjective nature of lyric quality and quantifying a 'good match' to an audio file complicates the nature of validating a model. Traditional metrics may not fully capture the essence of what makes lyrics resonate with listeners, due to the intricate interplay between words and music. Given the nuances of acoustic conditioned lyric generation, Lyricade seeks to synergize lyrics and music in a transformer-based finetuned GPT and address the aforementioned problems.

3 Related Work

The intersection of lyrics generation, signal processing, and transformer models constitutes an evolving field. Historically, the generation of melodies from lyrics has been prominent, focusing on semantic analysis and fitting it within musical compositions Lu and Eirinaki (2021). This process is somewhat inverse to our goal, which emphasizes generating lyrics influenced by underlying instrumental tracks. While GAN-based models have shown promise in this arena, particularly outperforming LSTM-based counterparts in song selection and generation, our exploration pivots towards utilizing GPT models for their generative capabilities and flexibility Lu and Eirinaki (2021).

Ding et al. (2024)'s SongComposer represents a significant stride in automated song creation, combining lyrical and melodic generation within a unified LLM framework that employs symbolic song representations Ding et al. (2024). While SongComposer extends to melody generation, its methodological foundations in feature extraction and model training offer valuable insights for our work, particularly in processing audio inputs to influence lyrical output.

In a similar vein, Melistas et al. (2021) generate lyrics and vocal melodies conditioned on instrumental tracks, employing a memory-efficient Transformer architecture for this sequence-to-sequence task Melistas et al. (2021). Their approach to decoupling lyrics from melody generation, allowing the integration of pretrained language models, mirrors our efforts in prompt engineering and feature weighting. Their methodology underscores the intricate relationship between instrumental accompaniment and vocal components—a dynamic we aim to capture by incorporating signal processing features for enhanced lyric generation.

In terms of previous work integrating signal processing into transformer models to process and generate audio content, LauraGPT (2023) handles audio and text inputs to output across modalities Chen et al. (2023). LauraGPT aligns closely with our objective of fusing lyrical data with processed audio features, demonstrating combinations of continuous and discrete audio features has informed our approach to integrating diverse acoustic characteristics into the lyric generation framework.

Moreover, advancements in audio-text Transformers, as discussed by Wang et al. (2023b) and Rubenstein et al. (2023), have influenced our strategy in representing audio signals Wang et al. (2023); Rubenstein et al. (2023). Opting for a Conformer-based encoder to transform audio inputs into continuous representations has been a pivotal decision, informed by the limitations of using a singular tokenizer for diverse audio tasks.

Recent contributions to the field further emphasize the complexity and multifaceted nature of lyrics generation. Zhang et al. (2024) delve into syllable-level constraints and semantic patterns essential for generating coherent lyrics aligned with melodic structures Zhang et al.. Meanwhile, Yuan et al. (2024)'s ChatMusician introduces an LLM with intrinsic musical capabilities, demonstrating that integrating musical understanding does not compromise, but rather enhances, linguistic generation capabilities Yuan et al. (2024).

Despite these developments, the integration of acoustics and signal processing with transformer-based models for lyrics generation remains a burgeoning area of research. Our project, Lyricade, contributes to this evolving landscape by harnessing the strengths of transformer models, enriched with acoustic features, to generate lyrics that resonate with the instrumental essence and artist's style.

While building on existing methods in literature, Lyricade is unique in our detailed acoustic characteristics extraction and feature weighting, along with dynamic generation and continous prompting. Unlike other models that might only broadly leverage audio features, Lyricade prioritizes tempo, energy, and valence. Moreover, Lyricade offers continuous generation tied to specific timestamps within a track, allowing for dynamic lyric creation with an adaptive feature weighting mechanism. This enables a customized generation process that pushes the boundaries of natural language processing for music applications.

4 Approach

4.1 Model Fine-tuning

We implement fine-tuning of the pre-existing GPT-2 model from the Hugging Face library, initially trained on a vast corpus of text data. We construct an extensive dataset of 1153 songs from the Spotify

API for song feature metrics and various open-source Kaggle datasets containing song lyrics. This fusion allows the GPT-2 to adapt the lyrical context of songs and their musical characteristics.

4.2 Data Integration and Tokenization

A significant challenge we encountered was the integration of numerical, categorical, and tokenized lyrics data into a cohesive training framework. Given that transformer models are designed to process string inputs, we developed a mechanism to standardize an input tensor structure of numerical and categorical data regarding acoustic features. Lyrics were often 20-50 lines which required tokenization in order to retain meaning across long strings. Through prompt engineering, we formatted input data to mirror the structured prompts e.g., "<VALENCE: 0.23> <TEMPO: 124> <ENERGY: 0.713>...". This approach ensures that each musical feature is distinctly recognized and utilized by the model during the training phase.

4.3 Acoustic Segmentation & Continuous Generation

To address the dynamic nature of songs, where different segments may convey varying acoustic features, we employed an audio segmentation technique. Songs were segmented to 10-15 second increments, allowing for iterative lyric generation tailored to each segment. By using the last line of lyrics generated for the preceding segment as a contextual prompt for the next, we ensure a coherent and seamless lyrical narrative throughout the song.

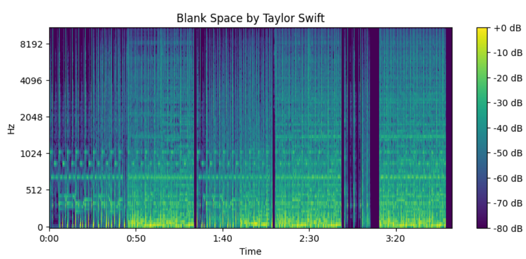


Figure 1: Mel Spectogram

Feature	Value
Duration (ms)	259370
Danceability	0.529
Energy	0.825
Key	4
Loudness	-4.454
Mode	1
Speechiness	0.0324
Acousticness	0.158
Instrumentalness	0.0131
Liveness	0.385
Valence	0.227
Tempo	93.969
Time Signature	4
Track Genre	pop

Figure 2: Audio Features: 'Blank Space' by Taylor Swift

4.4 Extraction and Scaling of Musical Features

A notable aspect of our approach is the extraction of musical features directly from MP3 files using the Librosa library McFee et al. (2015). To align these extracted features with the standardized values provided by the Spotify API, we scale them accordingly. Given the lack of transparency from Spotify regarding the computation of features like danceability and speechiness, we derive them from Mel spectrograms and other audio analysis techniques for approximation these features, ensuring our model's outputs are relevant and reflective of the input music's characteristics. These features include:

- Duration: Extracted using the Librosa library, the duration of the audio file is converted from seconds to milliseconds.
- Danceability: This complex feature is derived from the tempo, beat strength, rhythm stability, and overall regularity of the music. It is normalized to fall within a specified range to ensure consistency.
- Energy: Calculated as the mean root mean square (RMS) energy of the audio signal, this feature represents the intensity and activity level of the track.

- Key: The predominant key of the track is identified using the chroma feature across time, translating it into a numerical index that represents musical keys.
- Loudness: Similar to energy, loudness is determined by analyzing the RMS energy and normalized to reflect the track's average loudness level.
- Acousticness and Instrumentalness: These features are approximated by analyzing the spectral contrast and the presence of vocal frequencies, respectively, providing insights into the acoustic nature and instrumental content of the track.
- Liveness, Valence, and Tempo: These features are extracted using various Librosa functions and normalized. Liveness detects the presence of an audience in the recording, valence represents the musical positiveness, and tempo measures the speed of the track.

4.5 Feature Weighting

Recognizing the varying influence of each acoustic feature, we introduced a feature weighting mechanism for features in the numerical layer. Through our parameter fitting routine, minimizing training loss across features, we prioritized relevant characteristics to lyrics generation like tempo, energy, and loudness over danceability, for example. We ran parameter sweeps to manually optimize this on our initial version model and visualized our findings.

5 Experiments

5.1 Data

The lyrics dataframe is constructed by merging and normalizing data across various datasets, including Genius Song Lyrics with Language Information[1], Song Lyrics Dataset[2], and Lyrics Generation Dataset[3]. We used another Kaggle Exploring Spotify [4] features dataset with pre-extracted acoustic features resulting in a combined dataset of 1153 rows and 22 columns containing song metadata, acoustic features, and lyrics.

5.2 Model Iterations

Our model iterations reflected an evolving strategy in integrating acoustic features, with each version addressing unique challenges we noticed while training.
Version 1: Simple Unweighted This initial model served as a foundation for fine-tuning and parameter optimization, focusing on lyrics continuation. It struggled with incorporating numerical features effectively, largely acting as a basic continuation model that performed standardly when prompted with artist names or song topics. Notably, lyric encodings, numerical features and categorical features were stored as distinct tensors

Version 2: Simple Weighted Building on the first version, this iteration introduced feature weighting and bar-separated prompting for input values. Despite improvements, it essentially remained a lyrics completion model. The distinct numerical values didn't significantly affect output uniqueness relative to the input numeric prompts, indicating its efficacy in completion but not in aligning with the beat.

Version 3: Prompt Engineered and Weighted To better integrate numeric features, our final version concatenates numerical, categorical, and tokenized lyrics into a single string, structured to match a feature-prompting regex. It treats the combined data as one comprehensive tensor and applies feature weighting before this integration. This approach marks a substantial evolution from previous models, aiming to closely align the generated lyrics with the acoustic profile and instrumental essence of the input track.

[1] https://www.kaggle.com/datasets/carlosgdcj/genius-song-lyrics-with-language-information

[2] https://www.kaggle.com/datasets/deepshah16/song-lyrics-dataset

[3] https://www.kaggle.com/datasets/pratiksaha198/lyrics-generation?select=LYRICS_DATASET.csv

[4] https://www.kaggle.com/code/alankarmahajan/exploring-spotify-dataset

5.3 Evaluation method

We use BLEU score and ROGUE score as objective metrics to evaluate the performance of our model. To complement these objective evaluations, given the subjective nature of lyrics generation as a task, we conduct a qualitative assessment through survey interviews. Participants were presented three distinct models: a simple unweighted, a simple weighted, and a prompt engineered weighted.

5.4 Experimental details & Hyperparameter fitting

In the refinement of our model, extensive hyperparameter tuning and experimentation were conducted to optimize performance and address overfitting. Initially, we employed the GPT-2 model from Hugging Face's library, experimenting with various epochs to determine their impact on the model's behavior and loss metrics.

We tested 10, 50, and 300 epochs, where each epoch consisted of 132 batches, with training times initially averaging 45 seconds per epoch and later increasing to approximately 1.5 minutes for longer sessions. The experiments were conducted on our personal computer equipped with an NVIDIA GPU (RTX 4080).

At the outset, 5 epochs were tested for short-term training, yielding preliminary insights but resulting in suboptimal lyric coherence. The high average training loss of 1.763 was also indicative of this. Our decision to expanding to 300 epochs signficantly diminished the training loss (yielding an average of 0.2492 and lows of 0.0232). However, this model induced severe overfitting, as evidenced by the model's consistent behavior of simply copying exact lines from the songs of the training dataset during lyric generation, thus undermining generative robustness.

To counteract this, we integrated L2 regularization with a weight decay of 0.01 into our AdamW optimizer. Despite some improvements, the model remained excessively dependent on the training data, often still producing deterministic outputs. Because of this, we settled on a 50-epoch training regime, which struck an effective balance between model complexity and generalization, leading to a threefold decrease in the overfitting score and enhanced originality in lyric generation. Additionally, a 10-epoch run was conducted as a baseline for comparison.

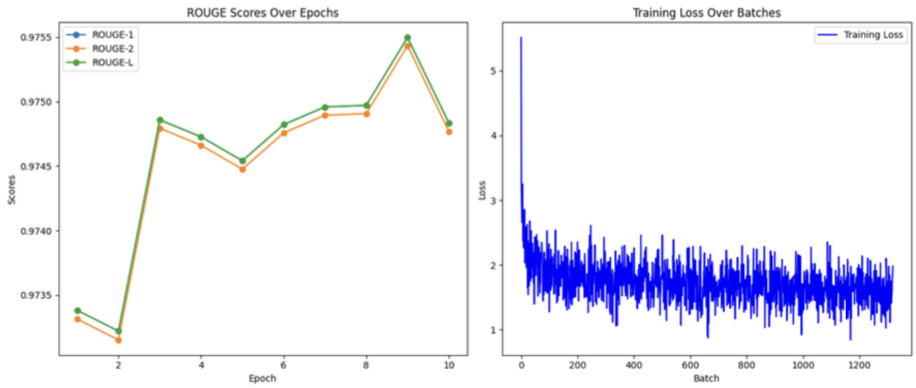
5.5 Feature Weight Exploration

In our investigation, we explored a wide array of parameter settings, initially testing 'high' and 'low' assessments for each parameter to determine its relative importance. We constructed 15 parameter sweeps manually although future work could automate a fitting routine. Our approach allowed us to identify a particularly effective set of feature weights:

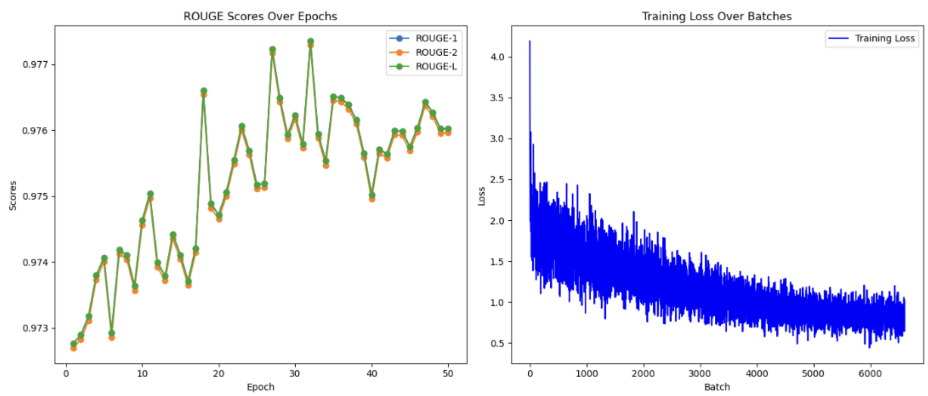
Feature	Weight
Tempo	4.5
Valence	4.0
Popularity	4.0
Danceability	3.5
Energy	4.5
Loudness	3.0
Speechiness	2.5
Acousticness	3.5
Instrumentalness	3.5
Liveness	3.0
Explicit	2.5

Table 1: Optimized Feature Weights for Model Training

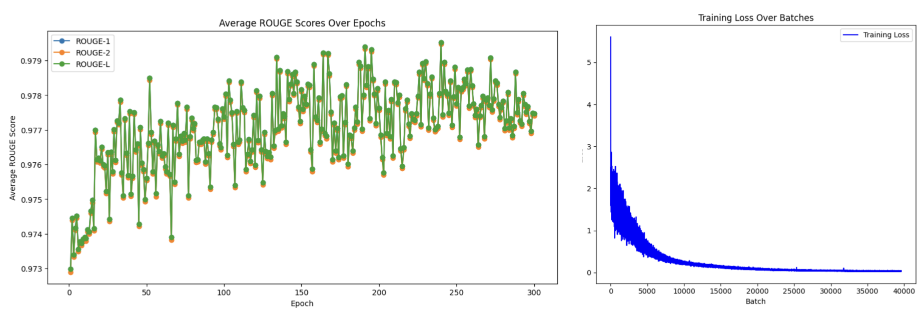
Our analyses demonstrated that the chosen feature weighting scheme (Feature Run #1) significantly outperformed others in terms of training loss, achieving a training loss of approximately 1.7 within just 5 epochs. While extending the number of epochs might have revealed greater differentiation, computational constraints limited our exploration. Nonetheless, the scaling of these optimal feature weights from 0-1 yielded the highest accuracy in terms of ROUGE-L and BLEU scores. For practical implementation in our acoustic feature extraction methodology, we scale our training set from 0-5.



(a) 10 Epoch Run



(b) Training Loss



(c) 300 Epochs

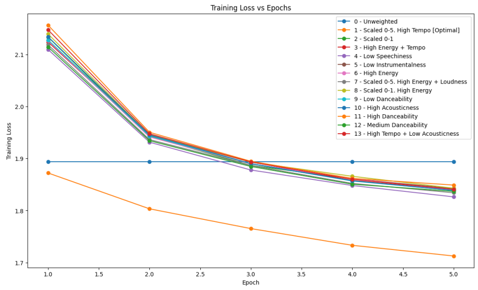
Figure 3: Hyperparameter Optimization: ROGUE scores and Training Loss

This scaling adjustment helps preserve the resolution of features, thereby addressing any disparities in model accuracy.

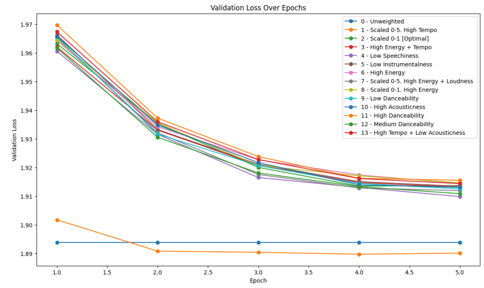
5.6 Results

Across epoch testing, we found that 50 epochs was within the "goldilocks" zone of our model. The 300 epoch run was extremely overfit, producing 25-30 lenght n-grams directly found in our original dataset. Naturally, the 10-epoch model, though less complex, inadequately captured detailed user inputs and demonstrated a higher training loss. For instance, when numerical features such as valence were drastically changed (in hopes of eliciting different lyrics), very little to change would occur.

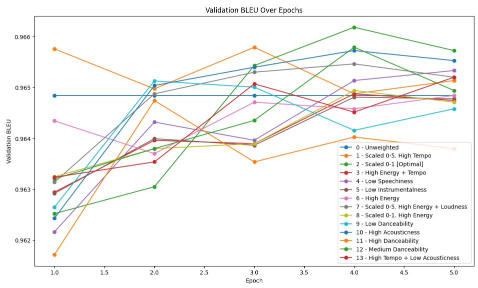
The detailed metrics for each training configuration are as follows:



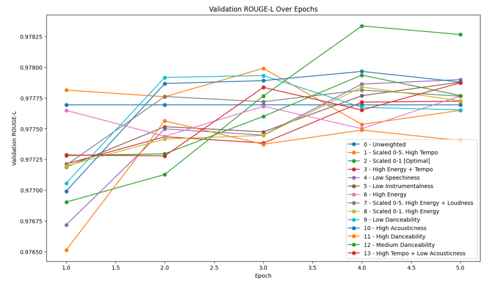
(a) Training Loss Across Epochs



(b) Validation Loss Across Epochs



(c) BLEU Score Across Epochs



(d) ROUGE Score Across Epochs

Figure 4: Feature Weight Sweeps for Optimization: Optimal Feature set in bottom orange line suggests best training loss and high accuracy scores within 5 epochs

- **10 Epochs:**
  - Average Training Loss: 1.7301, Final Loss: 1.9826
  - Validation BLEU Score: 0.9378
  - Average ROUGE Scores: -1: 0.9746, -2: 0.9745, -L: 0.9746
  - Overfitting Indicator: 0.1632
- **50 Epochs:**
  - Average Training Loss: 1.1620, Final Loss: 0.6099
  - Validation BLEU Score: 0.9446
  - Average ROUGE Scores: -1: 0.9765, -2: 0.9764, -L: 0.9765
  - Overfitting Indicator: 1.5049
- **300 Epochs:**
  - Average Training Loss: 0.2492, Final Loss: 0.0377
  - Validation BLEU Score: 0.9434
  - Average ROUGE Scores: -1: 0.9771, -2: 0.9770, -L: 0.9771
  - Overfitting Indicator: 4.0114

Hyperparameter fitting and experimental details highlight the importance of balancing training duration with model generalization. The 50-epoch configuration proved optimal, demonstrating significant reduction in overfitting and improved generativity in lyric production, as substantiated by both qualitative and quantitative analyses.

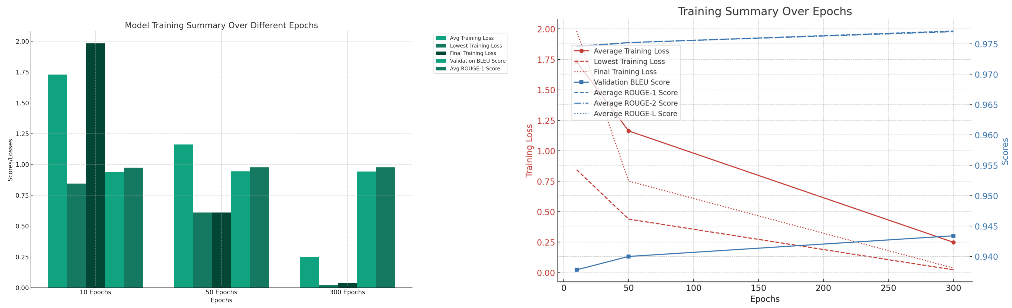


Figure 5: # of Epochs vs. Training Scores & Loss

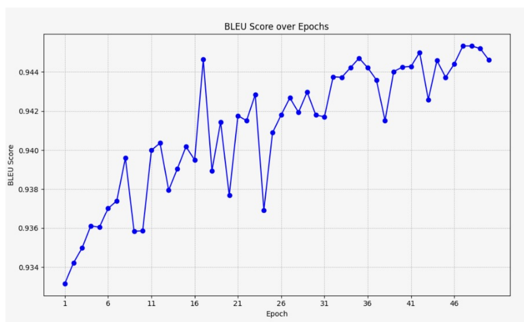


Figure 6: BLEU Score Across 50 Epochs

Generated Guitar Lyrics

Model Input:
| Artist - Charlie Puth | Genre - Guitar | Explicit - 0 |
(extracted from guitar.mp3)

Model Output:
"i don't know what to make of it but i can't help but wonder what it's like to be in love with someone who's not my type of person. i'm not sure what i'd do if i were you but you're the only one i've ever known if you were me then you'd be the one to tell me that you love me and i would be your lover. you know i love you i know you like it when you touch me. i want you to know that i'll be there for you. i wanna be with you and you'll never let me go. i won't let you go no matter how hard you try to keep me from loving you 'til the day i die"

6 Analysis

For the qualitative analysis, we evaluated the impact of our model iterations on the quality of generated lyrics through a blind survey of 25 individuals. Each was exposed to three songs across different genres, followed by three sets of lyrics corresponding to each of our model iterations - simple unweighted, simple weighted, and prompt-engineered weighted. After listening to the original tracks with lyrics, participants were presented with the instrumental versions alongside the three lyric options, enabling them to cast votes based on their preferences for best matched lyrics to audio 2.

Our analysis revealed a distinct preference for the prompt-engineered weighted model, especially in the context of indie genre lyrics. This preference suggests that the slower tempo characteristic of indie music, combined with the refined integration of tempo in the final model iteration, resonated more effectively with the survey participants. In contrast, the rap/hip-hop genre did not exhibit a

Table 2: Survey Results Summary for Lyrics Generation

Instrumental Track	Genre	Model	# Votes	% Votes
Blank Space (Taylor Swift)	Pop	Simple Unweighted	2	8%
		Simple Weighted	6	24%
		Prompt Engineered Weighted	17	68%
Another Late Night (Drake)	Hip-Hop	Simple Unweighted	7	28%
		Simple Weighted	8	32%
		Prompt Engineered Weighted	10	40%
Motion Sickness (Phoebe Bridgers)	Indie	Simple Unweighted	0	0%
		Simple Weighted	3	12%
		Prompt Engineered Weighted	22	88%

significant bias towards any particular model iteration, indicating that our model's improvements were less impactful in genres with a different set of musical and lyrical structures.

Furthermore, our feature weighting experiment findings sheds light on the acoustic characteristics that play pivotal roles in pairing lyrics with music. By fine-tuning the weights of various features, we discovered that tempo and energy are the most influential factors in determining lyric compatibility. On the other hand, features such as loudness, speechiness, and liveness were assigned lower weights, suggesting they have a lesser impact on the alignment between lyrics and their corresponding musical backdrops.

Our findings revealed consistently high BLEU and ROUGE scores, indicating a notable alignment between generated text and the reference text. However, as the training jumped from 50 to 300 epochs, we observed a convergence of BLEU and ROUGE scores, while the loss function continues to decrease. Additionally we observe the overfitting metric to indicate the model becoming increasing specialized to the training data. This underscores the importance of both monitoring the training and validation metrics. Our decision to select the 50-epoch model was driven to strike a balance between model complexity and generalization. Nonetheless, even with the 50-epoch model, we noted residual aspects of overfitting in instances where generated lyrics closely resembled lines from existing songs. To address this, we might consider careful selection and fine-tuning of regularization techniques and hyperparameters.

Our analysis also unveiled inconsistencies in features such as the explicit tag or censoring, which we attributed to disparities within our dataset's labeling and categorization. A remedy would be to implement further dataset processing before training or implementing a post-processing routine, where misrepresentations in outputs can be filtered. Additionally our survey underscored a positive reception toward rhyming lyrics among users, suggesting the incorporation of a rhyme metric in our training regimen as a potential enhancement avenue.

7 Conclusion

Lyricade tackles the challenge of integrating acoustic signal processing with transformer-based lyric generation, leading to the creation of a model capable of producing contextually relevant and musically coherent lyrics. We fine-tuned the GPT-2 model, leveraging instrumental tracks and contextual cues, achieving a training loss of 0.75, BLEU score accuracy of 0.94, and a ROUGE-L score of 0.9752. Our experiments revealed the delicate balance between feature weighting and hyperparameter optimization to prevent overfitting while maintaining generative creativity. In our testing, we identified 50 epochs as the optimal training length, although further refinement may enhance performance.

The primary limitations of our work include the constrained size and quality of our dataset, leading to occasional inaccuracies in content generation, such as genre misclassifications and explicit content mislabeling. Moreover, the model's dependence on instrumental MP3 files for lyric generation presented a challenge in achieving fully autonomous lyric generation.

Future Work

Looking ahead, we aim to:

- Acquire a larger, better-formatted dataset to enhance the model's understanding of various musical elements and improve the accuracy of generated content.
- Develop a more user-friendly interface, potentially through a full-fledged website, to facilitate easier interaction with the model.
- Enable the model to generate lyrics based solely on textual input specifying characteristics like tempo, energy, and desired genre, thereby eliminating the dependency on instrumental files.
- Improve the lyric output formatting to include structured verses, choruses, and clear sentence demarcations, enhancing the readability and aesthetic appeal of the generated lyrics.
- Continue refining the model to further reduce loss and improve the quality of generated lyrics, exploring new methods and technologies to enhance its performance and reliability.

In conclusion, Lyricade represents a significant step forward in the field of AI-driven creative writing, blending music and text in novel ways to generate engaging and meaningful lyrics. Our work opens the door to new possibilities in the realm of automated music production, with ample opportunities for future enhancements and research.

References

Qian Chen, Yunfei Chu, Zhifu Gao, Zerui Li, Kai Hu, Xiaohuan Zhou, Jin Xu, Ziyang Ma, Wen Wang, Siqi Zheng, et al. 2023. Lauragpt: Listen, attend, understand, and regenerate audio with gpt. *arXiv preprint arXiv:2310.04673*.

Shuangrui Ding, Zihan Liu, Xiaoyi Dong, Pan Zhang, Rui Qian, Conghui He, Dahua Lin, and Jiaqi Wang. 2024. Songcomposer: A large language model for lyric and melody composition in song generation. *arXiv preprint arXiv:2402.17645*.

Junlan Lu and Magdalini Eirinaki. 2021. Can a machine win a grammy? an evaluation of ai-generated song lyrics. In *2021 IEEE International Conference on Big Data (Big Data)*, pages 4896–4905. IEEE.

Brian McFee, Colin Raffel, Dawen Liang, Daniel PW Ellis, Matt McVicar, Eric Battenberg, and Oriol Nieto. 2015. librosa: Audio and music signal analysis in python. In *SciPy*, pages 18–24.

Thomas Melistas, Theodoros Giannakopoulos, and Georgios Paraskevopoulos. 2021. Lyrics and vocal melody generation conditioned on accompaniment. In *Proceedings of the Workshop on Natural Language Processing for Music and Audio (NLP4MusA)*, pages 17–25, Online. Association for Computational Linguistics.

Paul K Rubenstein, Chulayuth Asawaroengchai, Duc Dung Nguyen, Ankur Bapna, Zalán Borsos, Félix de Chaumont Quitry, Peter Chen, Dalia El Badawy, Wei Han, Eugene Kharitonov, et al. 2023. Audiopalm: A large language model that can speak and listen. *arXiv preprint arXiv:2306.12925*.

Xiaofei Wang, Manthan Thakker, Zhuo Chen, Naoyuki Kanda, Sefik Emre Eskimez, Sanyuan Chen, Min Tang, Shujie Liu, Jinyu Li, and Takuya Yoshioka. 2023. Speechx: Neural codec language model as a versatile speech transformer. *arXiv preprint arXiv:2308.06873*.

Ruibin Yuan, Hanfeng Lin, Yi Wang, Zeyue Tian, Shangda Wu, Tianhao Shen, Ge Zhang, Yuhang Wu, Cong Liu, Ziya Zhou, et al. 2024. Chatmusician: Understanding and generating music intrinsically with llm. *arXiv preprint arXiv:2402.16153*.

Zhe Zhang, Karol Lasocki, Yi Yu, and Atsuhiro Takasu. Syllable-level lyrics generation from melody exploiting character-level language model.

A Appendix (optional)

- See our Demo!
- See our Code!
- See Qualitative Survey!

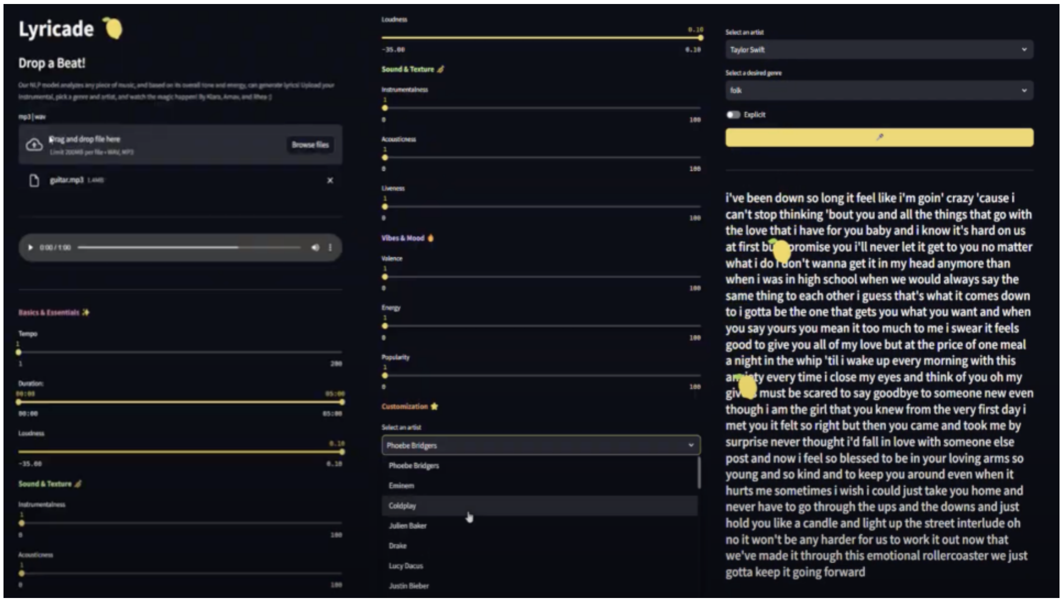


Figure 7: Front End of Lyricade: song input and lyrics generation

A.1 Front End

Our front end user interface offers a straightforward experience for users to upload MP3 tracks and manually adjust musical features as needed. With a simple drop-down menu for selecting an artist and genre, interacting with Lyricade becomes an effortless and smooth process.

A.2 More Output

Here are some more examples of produced lyrics:

RnB Lyrics

Model Input:
| Artist - Cardi B | Explicit - 1 | Genre - RnB |
(extracted from input.mp3)

Model Output:
"baby baby when we first met i never felt something so strong you were like my lover and all of a sudden you went and left. i didn't know how to follow it's like a shot that spun me around. oh no so let the music blast we gon' do our dance bring the doubters on they don't matter at all these rappers are like hunger games put a price on your head and if i lost i go to rehab then that's money you can't play me. i need a remedy been running this shit for the last few months got my mind made up i ain't going nowhere i'm just tryna roll it thirteen blunts in my cylinders fillin' up this daily planner gettin' busy 'bout your favorite rapper set and just load it up make a couple gs and chill the night up in the club"

Indie Pop Lyrics

Model Input:
| Artist - Phoebe Bridgers | Explicit - 0 | Genre - Indie Pop |
(extracted from input.mp3)

Model Output:
"it's been a long time coming and i can't wait to get back to you 'cause i really wanna be with you forever and ever i've been waiting for so long but i finally see the light at the end of the road i'll be coming home"

Country Lyrics

Model Input:
| Artist - Taylor Swift | Explicit - 0 | Genre - Country |
(extracted from input.mp3)

Model Output:
"i don't know how i got here uh how did i end up in this position im in it's like a rollercoaster ride all the highs and the lows just keep on going up and you never know when you're gonna dip or fall off of this hill and i know you've been hurt by someone else but i can't let that stop me from loving you even if it kills me so let me be who i am i just wanna feel you in my arms holding me close to give me strength to keep me going strong i want you to know that i will never let you down and if i fall i'll fall flat on my shattered heart"