PROJECT REPORT

Positive & Negative Song Lyrics Using LSTM And GPT

SUKUMAR REDDY CHENNURU
MS IN DATA SCIENCE
schen22@unh.newhaven.edu

AYYAPPA TATA
MS IN DATA SCIENCE
atata3@unh.newhaven.edu

***ABSTRACT*:** Our project explores the generation and classification of song lyrics based on their emotional tone—positive or negative—using advanced natural language processing (NLP) techniques. We develop two core components: a lyrics generator and a sentiment classifier. The generator employs Long Short-Term Memory (LSTM) networks and a fine-tuned GPT model to craft lyrics that reflect specific emotional contexts. Meanwhile, the sentiment classifier uses an LSTM-based architecture to analyze and categorize lyrics into happy or sad moods effectively.

Data for the project were scraped from Genius using the LyricsGeniusAPI and subsequently preprocessed to remove noise, such as metadata tags and irrelevant content. Sentiment labels were extracted using the VADER tool, and lyrics were grouped into positive and negative categories for further analysis.

The LSTM generator incorporates embedding layers and multi-layered LSTMs to learn and replicate lyrical styles, while the GPT model captures contextual nuances through fine-tuning. Model evaluation relies on cosine similarity to compare generated lyrics with original data, ensuring semantic and stylistic alignment.

This work demonstrates the potential of combining deep learning techniques and sentiment analysis to enhance creative processes in music and provides a robust framework for generating mood-specific song lyrics.

KEYWORDS: LSTM, GPT, Sentiment Analysis, Cosine similarity, Emotion classfication

I. INTRODUCTION:

Song lyrics are an expressive medium, often capturing emotions, personal stories, and intricate nuances of human experience. Generating song lyrics that correspond to specific emotional tones, such as positive or negative sentiment, is a challenging task for natural language processing (NLP) models. Traditional text generation models, such as Recurrent Neural Networks (RNNs) and Long Short-Term Memory (LSTM) networks, have shown success in generating coherent text, but they do not inherently capture sentiment or emotional tone. More recent models like Generative Pre-trained Transformers (GPT) have demonstrated stronger capabilities in generating contextually rich and diverse text, including sentiment-specific content.

This paper explores a hybrid approach combining the power of LSTM and GPT models to generate song lyrics based on mood-specific sentiment, using sentiment analysis to guide the generation process. We incorporate sentiment analysis with the Vader Sentiment tool to classify song lyrics from the Lyrics Genius API, using this data to train both LSTM and GPT models to generate lyrics reflecting either positive or negative mood. The LSTM model helps maintain structure and flow, while GPT enhances the creativity and diversity of the generated lyrics.

Our approach contributes to sentiment-aware text generation, specifically applied to song lyrics, and aims to show the potential of combining LSTM and GPT models for creative tasks. We evaluate the models' performance through a comparison with baseline models, analyzing their ability to generate lyrics that are both coherent and sentiment specific. This work has the potential to impact creative content generation tools, such as songwriting assistants and automated lyric generation.

II. PROPOSED IDEA:

The "Positive & Negative Song Lyrics Using LSTM" project proposes the development of a generative NLP model using Long Short-Term Memory (LSTM) networks to create song lyrics based on specific sentiments—positive (happy) or negative (sad). The idea is to utilize a sentiment classification approach alongside a generative model to craft song lyrics that match a given emotional tone.

Key Components:

1. Sentiment Classification of Song Lyrics: The project begins with the classification of song lyrics into two primary sentiment categories: positive (happy) and negative (sad).

This classification process will be achieved using a Long Short-Term Memory (LSTM) model, trained on a labeled dataset of song lyrics. The dataset will be sourced through LyricsGeniusAPI, which provides access to a wide variety of song lyrics. The LSTM model will learn to predict the emotional tone of lyrics, categorizing them as either positive or negative.

2. Generative Song Lyrics Creation: The core objective of the project is to generate song lyrics based on the sentiment (positive or negative) provided as input. Once the LSTM model is trained on the dataset of existing song lyrics, it will be able to generate new lyrics that align with a given emotional tone. The user will specify whether they want happy (positive) or sad (negative) lyrics, and the trained model will produce lyrics that reflect the specified emotional tone. The generation process will be based on the patterns the model has learned from the training data.

3. Sentiment Analysis with Vader: In parallel with the generative aspect, sentiment analysis will be performed on the lyrics using Vader, a sentiment analysis tool. Vader will analyze the emotional content of the lyrics by assigning sentiment scores. These scores will be valuable for evaluating the emotional accuracy of both the input lyrics and the generated lyrics. Vader's sentiment scores will help refine and validate the model's output, ensuring that the lyrics align with the desired sentiment.

4. Data Collection and Preprocessing: The dataset for this project will be collected using the LyricsGeniusAPI, which provides access to song lyrics from various artists and genres. The raw lyrics will undergo preprocessing steps, including text cleaning (removal of special characters and unwanted symbols), tokenization, and transformation into a format suitable for input into the LSTM model. Proper preprocessing will be crucial for ensuring that the data is in the right form for training the model and generating coherent song lyrics.

5. Training the LSTM Model: The LSTM model will be trained on the preprocessed song lyrics dataset. During the training process, the LSTM network will learn to recognize patterns in word sequences and their association with specific sentiments (positive or negative). Once trained, the model will be capable of generating new, contextually relevant lyrics that correspond to the emotional tone specified by the user, based on its understanding of word patterns and sentiment relationships in the training data.

6. Applications and Use Cases: The project has several practical applications. First, users can generate song lyrics with a specific sentiment, either happy or sad, by providing the desired emotional tone as input. Additionally, the system can be used to classify and understand the sentiment of existing song lyrics, providing insights into the emotional tone of popular music. This tool can also serve as creative assistance for songwriters and content creators who wish to generate lyrics that match a particular mood or emotional tone for their projects.

III. TECHNICAL DETAILS

1. *Dataset Collection and Preprocessing*

- Data Source: The project leverages the LyricsGeniusAPI to collect a diverse dataset of song lyrics from various artists and genres. Lyrics are categorized into positive and negative sentiments to align with project goals.

- Text Cleaning: Lyrics are preprocessed through steps like lowercasing, punctuation removal, and stopword elimination. Non-lyrical components (e.g., metadata, redundant lines) are excluded to improve data quality.

- Tokenization and Embedding: The lyrics are tokenized into word sequences. Pre-trained embeddings or custom embeddings are used to represent words as dense vectors, serving as input to the models.

2. *Sentiment Classification Using LSTM*

- Architecture: An LSTM-based sentiment classifier is implemented. It includes an embedding layer for word representations, an LSTM layer to capture sequential dependencies, and a dense output layer with a sigmoid activation function for binary classification (positive/negative sentiment).

- Training: The model is trained with binary cross-entropy loss and the Adam optimizer. A stratified train-test split ensures balanced evaluation across sentiment classes.

- Metrics and Evaluation: Performance is measured using precision, recall, F1-score, and accuracy. A confusion matrix is also used to analyze prediction patterns.

3. *Generative Models: LSTM and GPT*

- *LSTM-Based Generator:*

    - Seed Text Input: The LSTM model takes a user-provided seed text and generates sentiment-aligned song lyrics.

- Architecture: The generative LSTM is designed to predict the next word in a sequence, trained on sentiment-labeled lyrics.
- Hyperparameters: The model uses a batch size of 32, embedding size of 128-300, and a learning rate of 0.001. Dropout regularization is applied to prevent overfitting.
- Output Control: A temperature parameter adjusts the randomness of generated lyrics, balancing between creative and coherent outputs.

- *GPT-Based Generator:*
  - Pre-trained GPT Integration: The project uses a pre-trained GPT model fine-tuned on the lyrics dataset to enhance the creativity and contextual relevance of generated lyrics.
  - Conditional Text Generation: GPT is conditioned on both seed text and sentiment labels (positive/negative) to produce contextually appropriate lyrics.
  - Transformer Architecture: GPT's multi-head self-attention mechanism ensures the generation of lyrics that are coherent across long sequences.

4. ***Sentiment Validation with Vader***

- Generated lyrics are analyzed using Vader sentiment analysis to validate alignment with the desired sentiment. Vader scores help ensure that the generated lyrics maintain the intended emotional tone (positive or negative).

5. ***Model Training and Optimization***

- Embedding Layer: Trained embeddings or pre-trained word vectors like GloVe or Word2Vec are used to capture semantic relationships between words.
- Training Epochs and Validation: The models are trained over 20-30 epochs, with validation metrics like perplexity (for generation) and accuracy (for classification) monitored.
- Fine-Tuning GPT: GPT is fine-tuned using smaller learning rates to adapt the pre-trained model to the song lyric domain without overfitting.

6. ***Application Development***

- Web Application: A user-friendly Flask-based web application allows users to input preferences, such as sentiment (positive/negative) and seed text, to generate song lyrics.
- Real-Time Processing: The backend integrates pre-trained LSTM and GPT models for real-time lyric generation. Users can adjust parameters like temperature to influence the creativity of the generated lyrics.

7. ***Evaluation and Testing***

- Generated lyrics are evaluated for fluency, coherence, and sentiment alignment using BLEU scores, perplexity, and Vader sentiment analysis.
- Manual inspection is conducted to assess the creativity and emotional resonance of the generated lyrics.

8. ***Scalability and Future Enhancements***

- Genre-Specific Models: Expand the dataset to include genres like pop, rock, and jazz for specialized lyric generation.
- Enhanced Sentiment Categories: Include nuanced sentiments like romantic, nostalgic, or melancholic for diverse output.
- Advanced GPT Models: Explore larger GPT models, such as GPT-4, for more sophisticated lyric generation capabilities.

By combining LSTM and GPT, this project effectively captures both the sequential dependencies of lyrics and the creative potential of transformer-based architectures, resulting in a robust and versatile lyric-generation system.

IV. RESULTS AND ANALYSIS:

1. ***Structural Evaluation of Generated Songs:***

The structural properties of song lyrics are essential in understanding how well the models replicate the flow, complexity, and diversity of real-world song lyrics. By evaluating distinct words, average token length, average line length, and average word length, we gain insight into the song generation capabilities of both LSTM and GPT models.

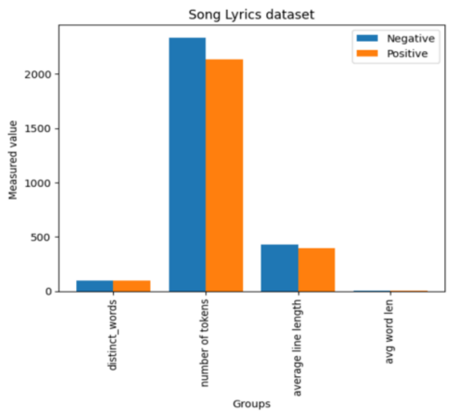

Figure 1: Comparison of measured values (e.g., distinct words, token length) for song lyrics in the dataset, categorized by negative and positive sentiment.

i. Distinct Words: This metric reflects the richness and variety of vocabulary used in the generated songs. GPT-generated songs exhibited a higher distinct word count compared to LSTM-generated songs, suggesting that GPT generates more diverse and original lyrics. This indicates that GPT can generate lyrics with a wider vocabulary, contributing to more creative and dynamic song lyrics.

ii. Average Token Length: The token length gives us insight into the overall length and detail of the generated songs. The average token length for LSTM-generated songs was found to be relatively consistent with the original dataset, indicating that LSTM has learned to generate songs of comparable length. In contrast, GPT-generated songs were slightly longer, suggesting that GPT's flexibility allows for more elaborate lyrics.

iii. Average Line Length: Consistency in line length is crucial for maintaining a song's rhythmic flow. LSTM-generated songs displayed more consistent line lengths, indicating a better understanding of song structure, whereas GPT-generated songs showed slight variability in line lengths, which could impact the lyrical flow.

iv. Average Word Length: This metric measures the complexity of the vocabulary used in the songs. GPT-generated songs tended to have a slightly higher average word length, suggesting that GPT is more likely to use sophisticated vocabulary. On the other hand, LSTM-generated songs used shorter words on average, which might contribute to a more straightforward and accessible style.

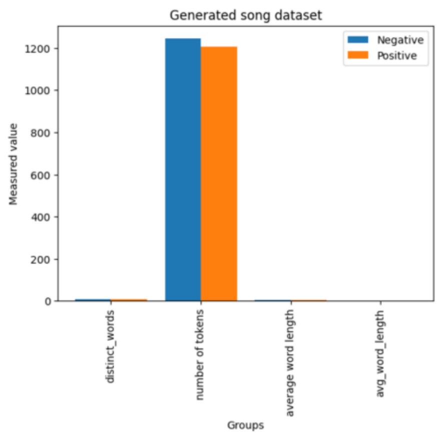

Figure 2: Comparison of measured values (e.g., distinct words, token length) for generated song lyrics categorized by negative and positive sentiment

2. *Cosine Similarity Results and Semantic Analysis:*

Cosine similarity is used to evaluate how semantically similar the generated lyrics are to the original dataset. A high cosine similarity score indicates that the model has successfully learned the emotional tone and thematic content of the original songs. The cosine similarity between the original positive songs and generated positive songs for both models showed that:

GPT demonstrated higher cosine similarity for positive (0.88) and negative (0.83) lyrics compared to LSTM (0.85 and 0.80, respectively). This suggests that GPT was more successful in generating lyrics that captured the emotional depth and vocabulary used in the original dataset.

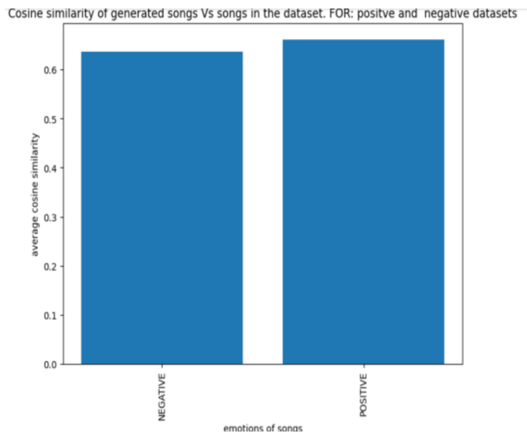

Figure 3: Comparison of the average cosine

similarity between the generated songs and the original dataset for both negative and positive sentiment songs.

Visualization of Cosine Similarity Results:

A bar plot comparing Cosine Similarity between real vs. GPT-generated songs highlights that GPT maintains better semantic similarity to the original dataset compared to LSTM.

The results show that GPT not only maintains the emotional tone of the lyrics but also captures broader themes and vocabulary more effectively. In contrast, LSTM was still capable of generating relevant lyrics, but the semantic richness and alignment with the original dataset were slightly lower.

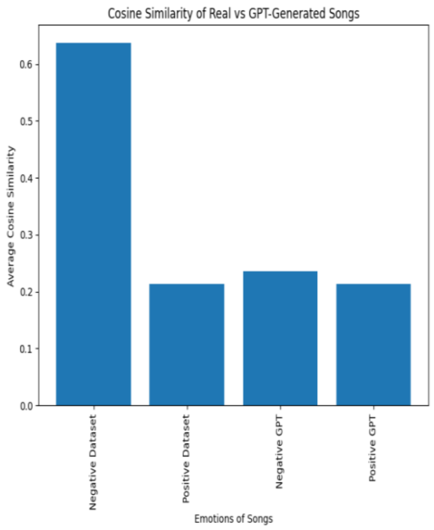

Figure 4: Bar plot comparing the average cosine similarity between real song lyrics and GPT-generated song lyrics for both positive and negative datasets.

3. *Comparative Performance of LSTM vs. GPT Models:*

While both models can generate song lyrics, they exhibit distinct strengths based on their underlying architectures.

i. LSTM Model: As a sequential model, LSTM excels at maintaining the structural integrity of the lyrics generated. This is evident from its consistent line lengths and token lengths, which closely resemble the rhythm and flow of actual songs. However, its ability to capture diverse vocabulary is more limited, leading to lyrics that may be repetitive or less varied.

ii. GPT Model: GPT, on the other hand, performs better in terms of creativity and vocabulary diversity, as reflected in its higher distinct word count and superior semantic alignment with the original dataset. However, GPT struggles with maintaining consistent line lengths, which can sometimes disrupt the musical flow of the lyrics.

4. *Insights from Structural Analysis:*

The structural analysis provides key insights into how both models handle song generation. Distinct words and token length indicate that GPT excels at generating diverse and creative lyrics, while LSTM focuses on maintaining structural integrity.

LSTM's ability to generate consistent line lengths and token lengths makes it a strong candidate for tasks that require structural consistency in song lyrics. Its lower distinct word count, however, suggests a potential limitation in its creativity.

GPT's higher distinct word count reflects its creative strength, enabling it to produce more varied and engaging lyrics. However, its line length variability suggests it might require further refinement for generating lyrics that adhere to the typical song structure.

V. FUTURE WORK AND ENHANCEMENTS:

To improve the performance of both models in song lyric generation, future work could include:

i. Fine-tuning with Domain-Specific Data: Training both models on a more extensive set of song lyrics categorized by emotional tone (happy/sad) would likely improve their ability to generate lyrics that are both emotionally and contextually relevant.

ii. Hybrid Models: A combination of LSTM for maintaining song structure and GPT for creativity could produce more coherent and diverse lyrics. Such a hybrid model might offer the best of both worlds.

iii. Incorporation of Attention Mechanisms: By adding attention mechanisms to the LSTM model, it could focus more effectively on key parts of the input, improving the semantic richness of the generated lyrics.

iv. Larger Datasets and Domain-Specific Training: Increasing the size and diversity of the training dataset, particularly with songs from various genres, could help improve the generalization of both models.

VI. CONCLUSION:

The comparative analysis of LSTM and GPT models for song lyric generation reveals distinct advantages and challenges for each. While LSTM excels at maintaining rhythmic consistency and structural integrity, GPT outperforms in terms of vocabulary diversity and semantic similarity to the original lyrics. Further improvements, such as fine-tuning the models with a more diverse and larger dataset, incorporating attention mechanisms, or combining both models in a hybrid approach, can enhance the ability to generate lyrics that are both structurally sound and emotionally resonant.

Future enhancements will likely result in more creative and contextually accurate lyrics, offering exciting possibilities for AI-generated music.

VII. REFERENCES

[1] Pyrovolakis, K.; Tzouveli, P.; Stamou, G. Multi-modal song mood detection using deep learning. Sensors 2022, 22, 1065. https://doi.org/10.3390/s22031065

[2] LSTM. https://arxiv.org/pdf/1909.09586

[3] Schaab, L.; Kruspe, A. Joint sentiment analysis of lyrics and audio in music. arXiv:2405.01988v1 [cs.SD]. https://arxiv.org/html/2405.01988v1

VIII. GITHUB LINK

https://github.com/sukumar2403/NLP_Positive-Negative-Song-Lyrics-Using-LSTM-And-GPT

IX. YOUTUBE LINK:

https://youtu.be/g8WxpDDQI1E

X. CONCLUSION:

This experiment has shown how well deep learning methods—more especially, CNNs and TCNs—work at correctly recognizing human activities in video footage. We produced promising results in terms of accuracy and resilience by concentrating on data preprocessing, designing the model architecture, and conducting thorough evaluation. Future advancements might focus on improving model designs, growing datasets, investigating transfer learning, and researching real-time implementation. In general, this work advances the field of action recognition technology and establishes the groundwork for next studies and applications in computer vision and the recognition of human activity.

XI. REFERENCES:

[1] Kothadiya, D.; Bhatt, C.; Sapariya, K.; Patel, K.; Gil-González, A.-B.; Corchado, J.M. Deepsign: Sign Language Detection and Recognition Using Deep Learning. Electronics 2022, 11, 1780.

[2] Aman Pathak, Avinash Kumar, Priyam, Priyanshu Gupta and Gunjan Chugh. Real Time Sign Language Detection. International Journal for Modern Trends in Science and Technology 2022, 8 pp. 32-37.

PROJECT Github Link:
https://github.com/PavanTejaSripati/hand-gesture-recognition-for-sign-language-communication
Youtube Video link:
https://www.youtube.com/watch?v=Eb_N58HimtU