COMP550 Project Report: Sentiment Analysis and Emotion Detection for Song Lyrics

Sagar Nandeshwar (Student ID: 260920948), Parham Ghasemloo Gheidari, Tanya Kumar

ABSTRACT

In this project, we implement two large CNN models for emotion recognition and sentiment analysis for song's lyrics. We collected the lyrics using a web scraper, and pre-processed them using several different techniques, and used them coupled with MoodyLyrics classification tags to train our models. We achieved a high accuracy and F1 scores for both emotion and sentiment analysis. Github:Lyrics Emotion Recognition

1 Intoduction

Music has always been one of the main forms of entertainment, while also having a high value as a source of communication. Messages conveyed with musical lyrics often have more impact, and depending on the harmonies that accompany the lyrics, create a varying range of emotions in an individual. That is why music has always been a powerful tool in many applications ranging from boosting morale for soldiers of war to calming visitors of a spa. While reading this, you could start thinking of a song that is or used to be stuck in your head for quite a while. While melodies alone can be interpreted differently and are prone to subjective analysis, lyrics often have a more generally agreed-upon interpretation, especially regarding the mood, vibes, or - simply put - emotions they make appear.

In recent years, similar to other forms of entertainment, music streaming services have been growing rapidly. With the COVID pandemic happening, all entertainment industries received a massive boost of engagement and music has not been an exception[4]. Most services such as Spotify or Apple Music, have implemented features to keep listening to music in a specific context and mood. These features have different names such as "Radio" or "Discover", but at their core, they all follow the same principle; Whether explicitly focused on a certain song or implicitly building upon the history of the user, they try to suggest songs that follow the same emotional patterns. This defines a challenge that despite having some solutions, is still far from being adequately addressed, with many users reporting their discontent with these features in social media such as Reddit.

With advances in the field of AI and specifically NLP, we are now able to approach the challenge defined earlier using language processing methods. As described in the first paragraph, using lyrics ensures a stronger confidence in detecting the overall emotions involved with a song. However, one of the attractions of music as a medium is its proneness to subjective interpretibility. So we have decided to focus on broader classification terms to address this challenge. We try to detect whether lyrics convey "positive" or "negative" sentiments and also classify the lyrics into four possible categories of emotion: "Happy", "Sad", "Angry", and "Relaxed".

Outline: Following this introduction, we first provide a brief mention of background and related works. Then we describe our data and its preparation steps. In sections 4 and 5, we explain our methodology followed by our results. We end this report with a discussion of the summaries of our findings and potential strengths and weaknesses of the proposed approach.

2 Background and Related Work

A paper titled "Music Emotion Classification based on Lyrics-Audio using Corpus-based Emotion" by Rachman et. al.[1] explores music emotion classification based on the integration of lyrics and audio features. This work employs Corpus-Based Emotion (CBE) to enhance the F-measure for emotion classification in musical documentation. Linguistic features, including psycholinguistic and stylistic aspects, are extracted from text-format lyrical data. Audio features are considered to encompass energy, temporal, and spectrum features derived from audio signal data. The research involves the use of the MIREX Dataset and emphasizes the importance of preprocessing and conversion due to the unstructured format of music documents. The authors conclude that the application of Random Forest methods yields the best test result for music emotion classification, achieving an F-Measure value of 56.8%.

In another work conducted at the University of China, Shi et. al. titled "Research on music emotion classification based on lyrics and audio."[2] follow the same objective by looking at both lyrics and audio. The authors mention that the limitations of single classification models were the reason behind the exploration of combining lyrics and audio classification for more effective emotion classification. The paper then proposes a hybrid approach, leveraging Latent Semantic Indexing (LSI) for lyrics classification, coupled with Support Vector Machines (SVMs) to handle dimensionality reduction and classification. In

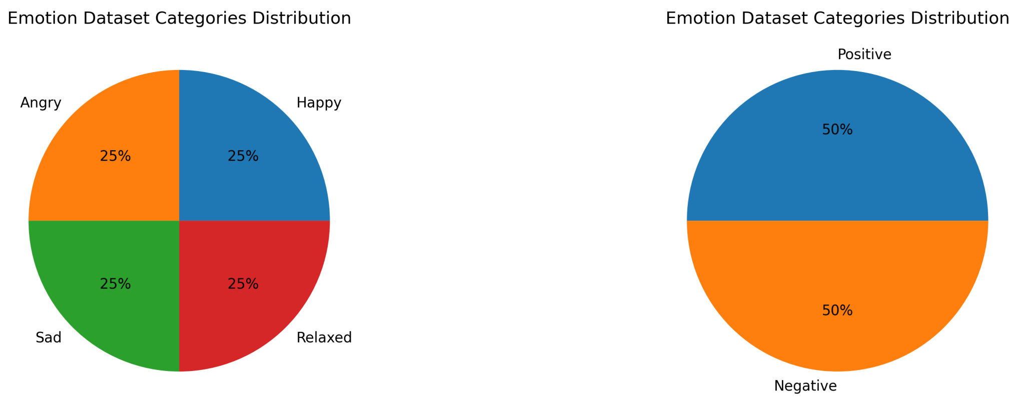

Figure 1. Dataset's emotions (left) and sentiment (right) categories distribution.

parallel, audio classification is performed by utilizing a Backpropagation (BP) neural network. The combination of the two classifications is achieved through the linear fractional stable motion (LFSM) algorithm. Experimental results demonstrate that this combined approach, incorporating multiple classification methods, yields higher accuracy in music emotion recognition compared to individual models.

The work titled "Lyrics-Based Emotion Classification Using Feature Selection by Partial Syntactic Analysis" by Kim et. al.[3] introduces a novel method for classifying emotions in songs solely based on lyrical content due to the limitations of approaches that only rely on melodical features. The proposed method uses feature selection through partial syntactic analysis, applying four syntactic analysis rules guided by an existing emotion ontology to extract emotional features from lyrics. The precision and recall rates for the emotion feature extraction are reported at 73% and 70%, respectively. The extracted emotion features are then combined with Naive Bayes (NB), Hidden Markov Model (HMM), and Support Vector Machine machine (SVM) learning methods, resulting in a maximum accuracy rate of 58.8%.

3 Data

We obtain our primary dataset from the MoodyLyrics[5] dataset which contains the song's name, artist, and emotion. However, we still needed to collect the lyrics of the songs that were included in MoodyLyrics. For this purpose, we implemented a web scraper that searches for lyrics on azlyrics.com and genius.com. We then collected the lyrics for roughly 2,000 songs and stored our data in two files, emotion.txt and sentiment.txt, to be used for emotion recognition and sentiment analysis respectively.

MoodyLyrics corpus of songs is a large dataset of roughly 2,000 songs, dividing them into four categories: Happy, Sad, Angry, and Relaxed. For the sentiment analysis, we define the "Happy" and "Relaxed" categories as "Positive" categories and "Sad" and " Anger" as "Negative". MoodyLyrics dataset has equal distribution of all 4 categories (see Fig. 1), hence this eliminates the biases that may be related to the size of the corpus.

Embedding: We used GloVe, which are Global Vectors for Word Representation. It help to map words into a meaningful space where the distance between words is related to semantic similarity.

Preprocessing: In the preprocessing phase, several important steps are employed to prepare the lyrics data for emotion classification. Tokenization is applied using *NLTK*'s *word_tokenize*, breaking down sentences into individual words, for instance, transforming "I like a cat" into ["I", "like", "a", "cat"]. Detokenization follows to convert lists of tokens back into continuous text strings, using NLTK's *TreebankWordDetokenizer*, ensuring the coherent representation of the processed data. Stop words, common but semantically less meaningful words like "and" or "the", are also removed using NLTK stopwords to improve the signal strength of the important words of the text. Punctuation is eliminated using Python's *string.punctuation*. Lemmatization is implemented with NLTK's *WordNetLemmatizer* to reduce words to their base forms (lemmas), aiming to standardize the vocabulary. Stemming is performed by utilizing NLTK's *PorterStemmer* and involves removing morphological affixes to retain only the word stems. Before being passed on to our model, lyrical text documents are converted into feature vectors using *sklearn*'s *CountVectorizer*, representing the data as a matrix of token counts, and normalization using sklearn's *TfidfTransformer* to transform the count matrix into a normalized TF (Term frequency) or TF-IDF (Term frequence-Inverse Document Frequency) representation. At the end of these preprocessing steps, we have prepared our raw lyrics data for downstream emotion classification analysis.

We performed a couple of small-scale experiments to decide what combination of these preprocessing steps is more beneficial for our purpose. We ended up using removal of stop-words, removal of punctuations, and lemmatizer, and avoided stemmer and POS tagging based on our experimental observations, as well as prior knowledge that we have acquired (most notably from the first and second programming assignments of the course).

4 Methods

Emotion Classification Model: For the emotion classification, we implemented a large neural network (NN) that consists of several layers, as listed below, in order:

- Dropout layer: Applies dropout to the input features. The Dropout layer randomly sets input units to 0 with a frequency of the rate at each step during training time, helping to prevent overfitting. We have selected the rate to be 0.2.
- Embedding layer: Turns positive integers (indexes) into dense vectors of fixed size.
- 1D convolution layers: This layer creates a convolution kernel that is convolved with the input layer over a single spatial (or temporal) dimension to produce a tensor of outputs. We used the ReLU (Rectified Linear Unit) activation function for 1D convolution layers.
- GlobalMaxPooling1D layer: Performs global max pooling operation for temporal data.
- Dense layers: 3 or 4 densely connected NN layers are the final layers of our NN. We again used the ReLU (Rectified Linear Unit) activation function for the Dense layers. However, for the last dense layer, we used the softmax function which has the same shape as the input.

We compiled the model with the Adam optimizer, which is a stochastic gradient descent (SGD) method that is based on adaptive estimation of first-order and second-order moments, and binary cross-entropy loss, which measures the difference between two probability distributions: the true distribution and the predicted distribution. We trained the model with 10 epochs and a batch size of 16.

Sentiment Classification: Similar to our emotion classification model, here we also used a neural network structure with several layers as listed before. For the last dense layer we used sigmoid function instead of softmax, in order to get binary output. The training was also performed by the Adam Optimizer. We trained the model with 10 epochs and a batch size of 16. The slight differences between our models - mainly in layer numbers - can be seen in Fig. 2. Further details are available in Fig. 3.

5 Results

Our Results can be seen in Figures 4 and 5. For emotion classification, our model can perform with an accuracy of %90 on the training set and %80 on the test set on the last epoch, which is higher than some of the related works that we have reviewed. From a critical point of view, this could be a result of not having enough data points (despite having almost 2000 songs) for our training, which could lead to problems such as overfitting and/or bias. However, we could also argue that our model is performing quite well because of the way we have structured our neural network. For sentiment analysis, our model performs well too, with an accuracy of higher than %95 on the training set and more than %90 on the test set. The same points that we made for our emotion classification model hold for this model as well. As for the F1 score, we get 0.80 for the emotion model and 0.93 for the sentiment model. From the confusion matrix for emotion classification, we observe that for most of the songs we accurately classify, the diagonals have high values. Similarly for sentiment models, we have high values in diagonal, confirming that the model has high accuracy.

6 Discussion

In this project, we implemented a large Convolutional Neural Network (CNN) model for emotion recognition and sentiment analysis for song lyrics. We collected the lyrics using a web scraper that we implemented ourselves and pre-processed the data for model training. In our results, we demonstrate that our models were capable of achieving high accuracies for both of the models. One potential reason behind this could be having a relatively large dataset of 2000 songs, compared to just 4 categories of emotions, for the emotion recognition model. The same principle also applies to our second model with only 2 categories of sentiments for sentiment analysis. The way that we have decided to construct our CNN network model also provides us with a

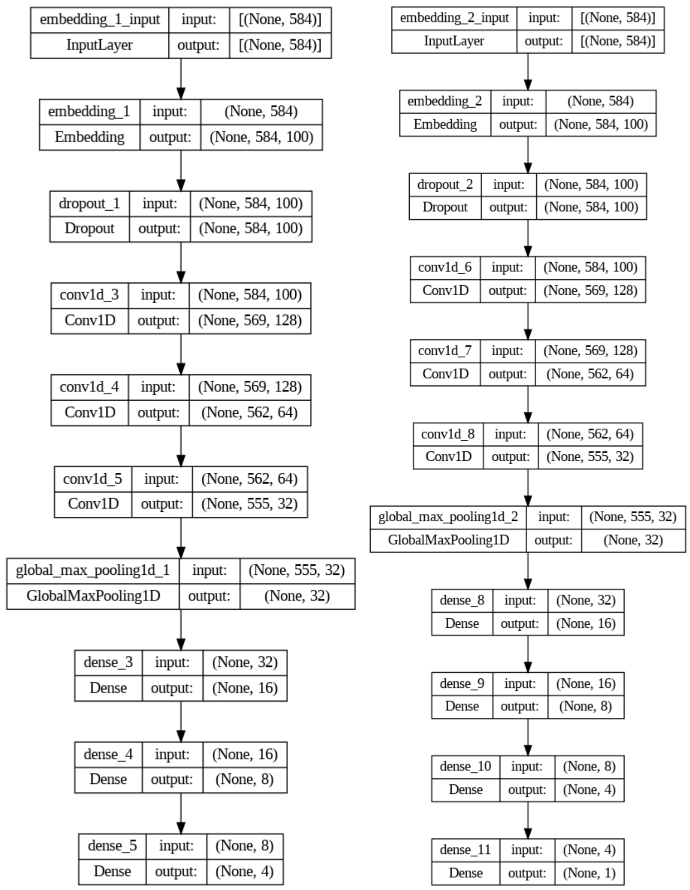

Figure 2. Model diagram for emotion classification (left) and sentiment classification (right)

```
Model: "sequential_1"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 embedding_1 (Embedding)     (None, 584, 100)          1389800

 dropout_1 (Dropout)         (None, 584, 100)          0

 conv1d_3 (Conv1D)           (None, 569, 128)          204928

 conv1d_4 (Conv1D)           (None, 562, 64)           65600

 conv1d_5 (Conv1D)           (None, 555, 32)           16416

 global_max_pooling1d_1 (Gl  (None, 32)                0
 obalMaxPooling1D)

 dense_3 (Dense)             (None, 16)                528

 dense_4 (Dense)             (None, 8)                 136

 dense_5 (Dense)             (None, 4)                 36

=================================================================
Total params: 1677444 (6.40 MB)
Trainable params: 287644 (1.10 MB)
Non-trainable params: 1389800 (5.30 MB)
_________________________________________________________________
```

```
Model: "sequential_2"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 embedding_2 (Embedding)     (None, 584, 100)          1389800

 dropout_2 (Dropout)         (None, 584, 100)          0

 conv1d_6 (Conv1D)           (None, 569, 128)          204928

 conv1d_7 (Conv1D)           (None, 562, 64)           65600

 conv1d_8 (Conv1D)           (None, 555, 32)           16416

 global_max_pooling1d_2 (Gl  (None, 32)                0
 obalMaxPooling1D)

 dense_8 (Dense)             (None, 16)                528

 dense_9 (Dense)             (None, 8)                 136

 dense_10 (Dense)            (None, 4)                 36

 dense_11 (Dense)            (None, 1)                 5

=================================================================
Total params: 1677449 (6.40 MB)
Trainable params: 287649 (1.10 MB)
Non-trainable params: 1389800 (5.30 MB)
_________________________________________________________________
```

Figure 3. Model details for emotion classification (left) and sentiment classification (right)

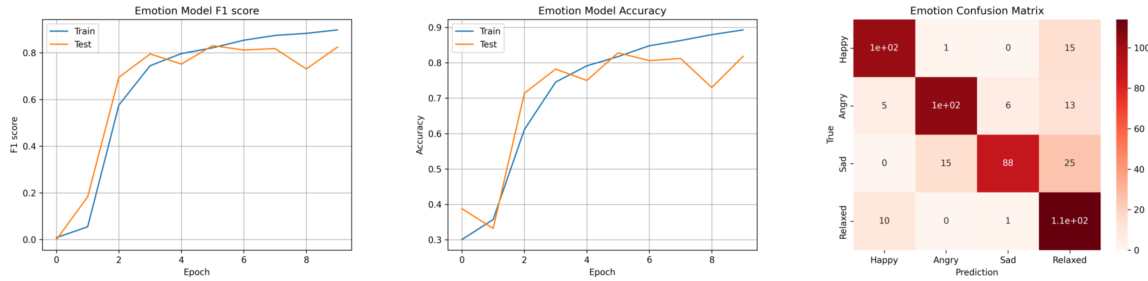

Figure 4. Evaluation metrics for emotion classification

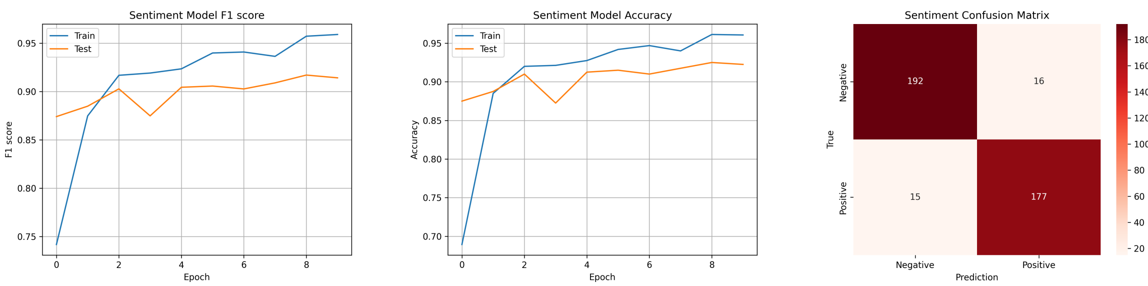

Figure 5. Evaluation metrics for sentiment classification

huge number of trainable parameters because of the sheer size of the network, which probably further helps in the analysis of the data and learning the patterns. Despite starting quite late and not having much time, we used a paid GPU to train our models in a feasible amount of time. Finally, we believe the preprocessing decisions that we have made - such as lemmatization and removing the stop words - have also played a key role in feature extraction, which has led to a significant performance of our models.

For future work, we were hoping for our model to incorporate the data about the artist. We believe that most of the artists have a genre that they follow regularly. This would be a piece of very beneficial information for both emotion and sentiment analysis.

7 Acknowledgements

Originally, we had a different topic proposed for our project that was focused on text summarization. Due to the unforeseen medical mishap that befell our teammate, Tanya, we had to move on to this topic which was discussed earlier in our group conversations during the proposal period. Since Tanya played a pivotal role in the text summarization project that we originally had in mind, we were left with no choice but to devise a plan b in case she wouldn't make a recovery in time. Despite all these events, hereby we acknowledge that everyone participated in each step of the project, communication was of high quality, and the work division was fair.

8 Involvement

Sagar Nandeshwar: Implemented web scapper, collected, implemented both CNN models for emotion recognition and sentiment analysis, model evaluation, produce graph and data, and wrote report. **Parham Ghasemloo Gheidari**: Wrote report and searched for resources and relevant papers. **Tanya Kumar**: Wrote project proposal

References

1. Rachman, Fika Hastarita, Riyanarto Sarno, and Chastine Fatichah. "Music Emotion Classification based on Lyrics-Audio using Corpus-based Emotion." International Journal of Electrical & Computer Engineering (2088-8708) 8.3 (2018).

2. Shi, Wanglei, and Shuang Feng. "Research on music emotion classification based on lyrics and audio." 2018 IEEE 3rd Advanced Information Technology, Electronic and Automation Control Conference (IAEAC). IEEE, 2018.

3. Kim, Minho, and Hyuk-Chul Kwon. "Lyrics-based emotion classification using feature selection by partial syntactic analysis." 2011 IEEE 23rd International Conference on Tools with Artificial Intelligence. IEEE, 2011.
4. Denk, Janis, et al. "The impact of COVID-19 on music consumption and music spending." Plos one 17.5 (2022): e0267640.
5. Çano, Erion, and Maurizio Morisio. "Moodylyrics: A sentiment annotated lyrics dataset." Proceedings of the 2017 international conference on intelligent systems, metaheuristics and Swarm intelligence. 2017.