EXPLORING AND APPLYING AUDIO-BASED SENTIMENT ANALYSIS IN MUSIC

Etash Jhanji*

Abstract

Sentiment analysis is a continuously explored area of text processing that deals with the computational analysis of opinions, sentiments, and subjectivity of text. However, this idea is not limited to text and speech, in fact, it could be applied to other modalities. In reality, humans do not express themselves in text as deeply as they do in music. The ability of a computational model to interpret musical emotions is largely unexplored and could have implications and uses in therapy and musical queuing. In this paper, two individual tasks are addressed. This study seeks to (1) predict the emotion of a musical clip over time and (2) determine the next emotion value after the music in a time series to ensure seamless transitions. Utilizing data from the Emotions in Music Database, which contains clips of songs selected from the Free Music Archive annotated with levels of valence and arousal as reported on Russel's circumplex model of affect by multiple volunteers, models are trained for both tasks. Overall, the performance of these models reflected that they were able to perform the tasks they were designed for effectively and accurately.

INTRODUCTION

Although sentiment analysis is well researched, studies focus on text-based emotion and represent emotions categorically.[6] People often express strong emotions through speech and audio, with more emphasis than in text. Other works focus largely on speech sentiment[10], not necessarily incorporating other sounds in an audio file. A computational model to interpret musical emotions can have medical uses, including therapeutic purposes, and also commercial uses in queueing music for streaming services. This project seeks to explore the capability of Long Short-Term Memory (LSTM) models to (1) predict the emotion of a musical clip over time and (2) determine the next "emotion value" after in a time series to ensure seamless transitions between audio sentiments.

BACKGROUND

Representing Emotion

Typically, text-based sentiment analysis attempts to classify text into categories of basic emotions as is supported by neuroscience studies on distinct neural circuits for emotions.[13] On the other hand, Russel's circumplex model of affect proposes that emotions arise as a product of two separate circuits: arousal and valence. These can also be renamed as activation and pleasantness, respectively.[8] This redefines the sentiment analysis task as a regression task with two outputs rather than a classification task. The database used also utilizes Russel's model on audio samples as the samples are annotated to show the emotion intended to be induced by the clip. Furthermore, the model can be divided into regions to represent emotions with more precision and can additionally describe intensity/severity as can be seen in Fig. 1. This is useful in the applications of audio analysis and provides a quantitative measure of emotion.

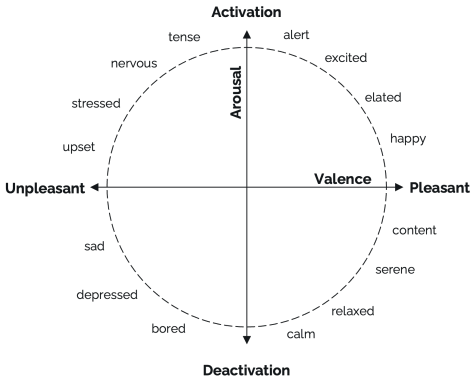

Figure 1: Russel's Circumplex Model of Affect

Audio Processing

There are many ways to represent audio files for input to a model, but this study uses Logarithmic Mel Spectrograms. This works by taking the raw audio waveform, applying a Fourier Transform, creating a spectrogram of the frequencies, applying the Mel scale to represent human frequencies better, and applying a logarithmic transformation.[4] Random Gaussian noise is also added to the Mel Spectrogram to improve the robustness of the model. The audio clips were spliced into half-second clips, sampled at 44100 Hz, and the Mel spectrogram has 128 frequency bins, an FFT length of 512, and a hop length of 2048 with Librosa. The output was a NumPy array or PyTorch tensor with a shape of 128 by 44 for 128 bins and 44 timesteps. Alternatively, the model could have used Mel Frequency Cepstral Coefficients (MFCCs) which tend to represent speaker data well[2], however, the model should not have to depend on speech features and should instead be able to understand purely instrumentals tracks as well. The extracted Mel Spectrogram dataset was then tied to its continuous arousal and valence coordinates. The full pipeline is demonstrated in Fig. 2.

Long Short-Term Memory Models

LSTM models are a subset of recurrent neural networks (RNNs). These models have a hidden state encoding previous information to keep a stateful memory of previous

* etashjhanji@gmail.com
Fox Chapel Area High School, Pittsburgh PA USA

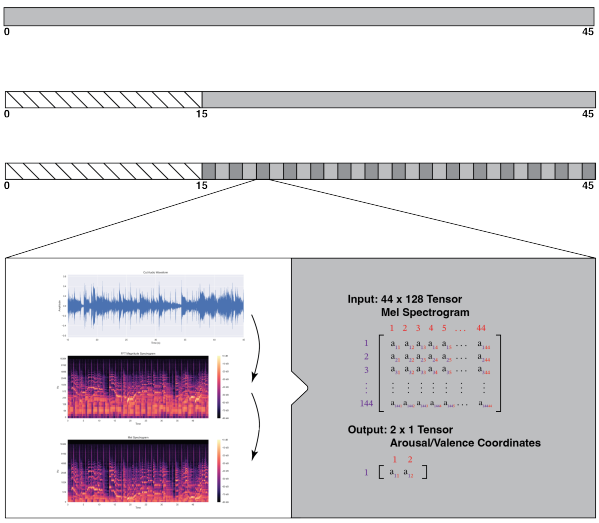

Figure 2: The full pipeline of audio processing including clipping, mel spectrogram, and storage format.

entries to the model. Traditional RNNs have a vanishing gradient problem where previous inputs are forgotten over time, but LSTMs solve this problem with a more sophisticated architecture. This type of model is able to outperform traditional models where long-term dependencies play a part in prediction, such as in audio where data takes place over time.

Dataset[1],[11]

The Emotion in Music Database (EmoMusic/1000 songs) consists of about 700 free-use songs that have been annotated by volunteers. The crowdsourced data consists of annotations sampled at 2 Hz (every 0.5 seconds) for arousal and valence of a clip on a scale from -1 to 1 each. The continuously sampled data points are the average of 10 volunteers' annotations at that instant to ensure stability of data. The standard deviations of the continuous annotations are also provided, averaging about 0.35 across all samples. This averaged standard deviation is the goal for the first model's root mean square error (RMSE) because it means the model's variation can be accounted for by the human variation in the dataset. The first 15 seconds of annotations in the data were dropped due to instability and the 45-second clips of audio are provided with the dataset.

TASK 1: PREDICTING EMOTION

Input/Output

The first task is to predict the arousal and valence values of a 0.5-second clip of audio's mel spectrogram. The input is a tensor of shape 128 by 44 representing the clips' mel spectrogram with specific settings (described in the background). The output will be a two-value tensor representing arousal and valence.

Evaluation Metric

The model will use mean squared error (MSE) for its evaluation metric and loss function. MSE penalizes heavily for deviation as it the square of the output unit. The target for model loss is 0.09 (or RMSE of 0.3) as described in the background section. Since human annotations vary by about 0.3 on average on the same songs, the error produced by the model can be accounted for by human error in the dataset.

Hyperparameter Optimization

This model uses the Adam optimizer. Experimenting with hyperparameters, the learning rate had the highest influence on results, and a low number of hidden layers performed best. To determine the number of epochs an early stopping technique was used in which the model automatically ended training after loss converged and stopped decreasing. A fixed batch size of 58 samples was used, where each batch contained every clip from one audio file. The best results found are shown in Table. 1 and Fig. 3. Although decreasing the learning rate from 5×10^{-5} may slightly improve the convergence of MSE, the final MSE is very similar whereas training time and epochs increase.

Table 1: Task 1 Best Results

Parameter	**Value**
Learning Rate	5×10^{-5}
Hidden Size	20
No. of Modules[1]	2
No. of Layers (per module)	2
No. of Layers (total)	4
Dropout Probability	0.1
Epochs	67
Train MSE	0.044
Validation MSE	0.054

Figure 3: Loss graphs (MSE) for training and validation varied with number of epochs for task 1 using the most optimal hyperparameter found. Shows convergence for training but slight possible overfitting.

[1] Where a "module" is an nn.LSTM object in the model and layers within a module are stacked

TASK 2: INTELLIGENT QUEUING

Input/Output

The second task is to predict the next pair of arousal and valence values from an arbitrarily chosen time sequence of length 10. The input is a tensor of shape 2 by 10 representing the arousal and valence over ten timesteps. The output will be a two-value tensor representing the next (11th) arousal and valence point.

Evaluation Metric

The model will also use MSE for its evaluation metric and loss function for the same reasons as in Task 1. The target MSE/RMSE are similar due to natural variation in humans but are expected to be lower in a simpler task.

Hyperparameter Optimization

Similar to the previous task, hyperparameter optimization was largely experimental with learning rate and epoch count having a high influence on results. This model still uses the Adam optimizer and mean squared error, however, batching is randomized and the task in simpler. The best results found are shown in Table. 2 and Fig. 4.

Table 2: Task 2 Best Results

Parameter	**Value**
Learning Rate	1×10^{-4}
Hidden Size	32
Batch Size	64
No. of Modules[2]	1
No. of Layers (per module)	2
No. of Layers (total)	42
Dropout Probability	0
Epochs	10
Train MSE	4×10^{-4}
Validation MSE	5×10^{-4}

Figure 4: Loss graphs (MSE) for training and validation varied with number of epochs for task 2 using the most optimal hyperparameter found. The loss converges very quickly after the first two epochs.

[2] Where a "module" is an nn.LSTM object in the model and layers within a module are stacked

Linear Regression Approach

Since the LSTM results showed that the task may have been too simple and revealed a possible exploding gradient issue, instead a linear regression approach was used. One in which arousal was the dependent variable and the other in which valence was the dependent variable with both having time as the independent variable. When the arousal and valence are plotted against time, they appear linear/constant (Fig. 5), but, zooming in, some clips show erratic, nonlinear behavior (Fig. 6). The linear regression showed that the data's overall trend may be able to be modeled by a regression, however, the model was not suitable for predicting exact values, especially within the songs (Fig. 7).

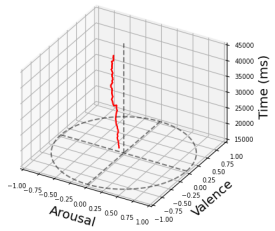

Figure 5: Arousal and valence for one song plotted against time on the z-axis. Dotted lines represent the centerlines and edges of the circumplex model.

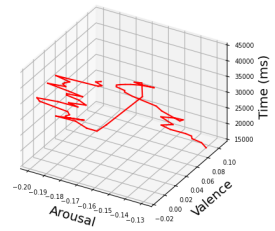

Figure 6: A zoomed view of Figure 5 that shows more erratic and nonlinear patterns in arousal and valence.

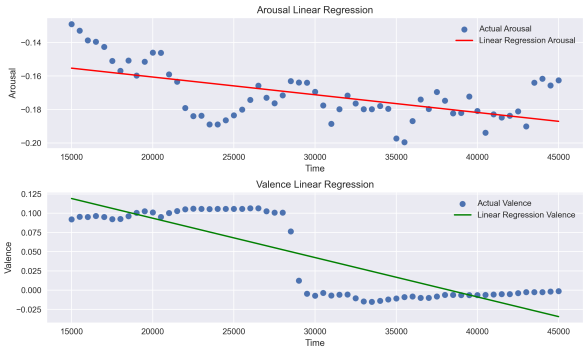

Figure 7: The linear regression results of one particular song. Shows a good model of the trend but a clear lack of accuracy around the center of the time axis.

RESULTS AND CONCLUSIONS

Task 1

The emotion prediction model got an MSE loss of about 0.055 in validation and 0.044 in training meaning that RMSE are 0.235 and 0.21, respectively. These RMSE values can be accounted for by the natural variation in even human choice annotation of this data, indicating the model is doing well, matching human predictions.

Task 2

The "next value" predictor model showed an MSE of about 0.0004 in training and 0.0005 in validation. These results indicate that the model performed well. The linear regression model proved to be less effective in predicting exact values but may be effective in considering general trends.

DISCUSSION

The ability to automatically determine quantitative values to describe emotion of a music clip can have uses in the real world. In the medical field, music is well known as having psychological effects on mood but has also been found to have effects on other neurological disorders including multiple sclerosis and Parkinson's.[9] Commercial applications of this software also include intelligent queueing of music to maintain a smooth flow of mood in music playback. A demonstrative implementation of this model is included in the public repository.[3]

FUTURE DIRECTION

For task 1 a broad, automated hyperparameter search may be able to refine and optimize hyperparameter for even better regression results. Different batching strategies and model complexities may also result in an overall superior model. The second task would benefit from many of the improvements of task 1, as well as using a statistic to objectively measure the performance of different models such as the linear regression. Adding variable length time-sequence input is also a possibility. Implementing this code in an open-source library that can be used in applications could greatly help those who use it. The code could also be implemented alongside the queuing algorithms of major streaming services (e.g. Spotify) to offer improved experiences to users.

ERROR ANALYSIS

Limited hyperparameter scope, input format, noise added, and human error could have resulted in errors in both task one and two's LSTM models. In model one it is important to note the slight overfitting occurring indicating that hyperparameters could likely be further optimized and in model two there may be an exploding gradient problem. In the second model, the predicted value tended to hold the arousal and valence constant due to the mostly constant nature of data, however, in songs with greater changes, the model reflected those changes. In implementation, adding a tolerance to allow for changes between songs would avoid holding the arousal and valence constant unintentionally.

[3] Please reference the GitHub repository for more information in the *demo.py* file

DATA AVAILABILITY

The Emotion in Music Database (1000 songs) dataset is available online after filling out a request form.[1],[11]

CODE AVAILABILITY

The code, models, and results from this dataset have been open-sourced under the MIT license and are available at `https://github.com/etashj/Exploring-and-Applying-Audio-Based-Sentiment-Analysis`.

ACKNOWLEDGMENTS

I would like to thank the developers of the Emotions in Music Database[1],[11] for making their code open source and free to use as it was of great importance to the development of this project. I would also like to thank the developers of the PyTorch[7], NumPy[3], pandas[12], and librosa[5] libraries for the Python programming language who also made their code open source which provided the framework for this project.

REFERENCES

[1] Florian Eyben et al. "Recent Developments in openSMILE, the Munich Open-source Multimedia Feature Extractor." In: *Proceedings of the 21st ACM International Conference on Multimedia*. MM '13. Barcelona, Spain: ACM, 2013, pp. 835–838. ISBN: 978-1-4503-2404-5. DOI: 10.1145/2502081.2502224. URL: http://doi.acm.org/10.1145/2502081.2502224.

[2] Todor Ganchev, Nikos Fakotakis, and Kokkinakis George. "Comparative evaluation of various MFCC implementations on the speaker verification task." In: *Proceedings of the SPECOM* 1 (Jan. 2005).

[3] Charles R. Harris et al. "Array programming with NumPy." In: *Nature* 585.7825 (Sept. 2020), pp. 357–362. DOI: 10.1038/s41586-020-2649-2. URL: https://doi.org/10.1038/s41586-020-2649-2.

[4] Bochen Li. *Machine Learning for Audio Signals: ECE 272/472 Audio Signal Processing*. URL: https://hajim.rochester.edu/ece/sites/zduan/teaching/ece472/lectures/Lecture_MachineLearningForAudio.pdf.

[5] Brian McFee et al. "librosa: Audio and Music Signal Analysis in Python." In: *SciPy*. 2015. URL: https://api.semanticscholar.org/CorpusID:33504.

[6] Walaa Medhat, Ahmed Hassan, and Hoda Korashy. "Sentiment analysis algorithms and applications: A survey." In: *Ain Shams Engineering Journal* 5.4 (2014), pp. 1093–1113. ISSN: 2090-4479. DOI: https://doi.org/10.1016/j.asej.2014.04.011. URL: https://www.sciencedirect.com/science/article/pii/S2090447914000550.

[7] Adam Paszke et al. "PyTorch: An Imperative Style, High-Performance Deep Learning Library." In: *Neural Information Processing Systems*. 2019. URL: https://api.semanticscholar.org/CorpusID:202786778.

[8] BRADLEY S. PETERSON, JONATHAN POSNER, and JAMES A. RUSSELL. "The circumplex model of affect: An integrative approach to affective neuroscience, cognitive development, and psychopathology." In: *Development and Psychopathology* 17.3 (2005), pp. 715–734. DOI: DOI:10.1017/S0954579405050340. URL: https://www.cambridge.org/core/product/9CC3D0529BCFA03A4C116FD91918D06B.

[9] Alfredo Raglio et al. "Effects of music and music therapy on mood in neurological patients." en. In: *World J Psychiatry* 5.1 (Mar. 2015), pp. 68–78.

[10] Maghilnan S and Rajesh Muthu. "Sentiment Analysis on Speaker Specific Speech Data." In: June 2017. DOI: 10.1109/I2C2.2017.8321795.

[11] Mohammad Soleymani et al. "1000 Songs for Emotional Analysis of Music." In: *Proceedings of the 2Nd ACM International Workshop on Crowdsourcing for Multimedia*. CrowdMM '13. Barcelona, Spain: ACM, 2013, pp. 1–6. ISBN: 978-1-4503-2396-3. DOI: 10.1145/2506364.2506365. URL: http://doi.acm.org/10.1145/2506364.2506365.

[12] The pandas development team. *pandas-dev/pandas: Pandas*. Version v2.1.0. Aug. 2023. DOI: 10.5281/zenodo.8301632. URL: https://doi.org/10.5281/zenodo.8301632.

[13] Christine D. Wilson-Mendenhall, Lisa Feldman Barrett, and Lawrence W. Barsalou. "Neural Evidence That Human Emotions Share Core Affective Properties." In: *Psychological Science* 24.6 (2013). PMID: 23603916, pp. 947–956. DOI: 10.1177/0956797612464242. eprint: https://doi.org/10.1177/0956797612464242. URL: https://doi.org/10.1177/0956797612464242.