Spring 5-24-2019

MUSIC MOOD CLASSIFICATION USING CONVOLUTIONAL NEURAL NETWORKS

Revanth Akella
San Jose State University

MUSIC MOOD CLASSIFICATION USING CONVOLUTIONAL NEURAL NETWORKS

Presented to

The Faculty of the Department of Computer Science

San Jose State University

In Partial Fulfillment

Of the Requirements of

Degree Master of Science

By

Revanth Akella

May, 2019

The Designated Project Committee Approves the Project Titled

MUSIC MOOD CLASSIFICATION USING CONVOLUTIONAL NEURAL NETWORKS

by

Revanth Akella

APPROVED FOR THE DEPARTMENT OF COMPUTER SCIENCE

San José State University

May 2019

Dr. Teng Moh Department of Computer Science

Dr. Leonard Wesley Department of Computer Science

Dr. Chris Pollett Department of Computer Science

ABSTRACT

Music Mood Classification using Convolutional Neural Networks

By Revanth Akella

Grouping music into moods is useful as music is migrating from to online streaming services as it can help in recommendations. To establish the connection between music and mood we develop an end-to-end, open source approach for mood classification using lyrics. We develop a pipeline for tag extraction, lyric extraction, and establishing classification models for classifying music into moods. We investigate techniques to classify music into moods using lyrics and audio features. Using various natural language processing methods with machine learning and deep learning we perform a comparative study across different classification and mood models. The results infer that features from natural language processing are a valuable information source for mood classification. We use methods such as term-frequency/inverse-document frequency, continuous bag of words, distributed bag of words and pre-trained word embeddings to connect lyrical features to mood classes. Different arrangements of the mood labels for music are explored and compared. We establish that features from lyrics with natural language processing methods demonstrate high levels of accuracy using CNNs. Our final model achieves an accuracy of 71% compared to existing methods using SVMs that achieve and accuracy of 60%.

Keywords—Deep learning, lyric extraction, mood classification, music classification, natural language processing, tag extraction

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

INTRODUCTION

The advent of audio streaming services has increased the accessibility to music. The fast growth of streaming services calls for a system of categorizing music based on listening habits and taste and mood categorization in music is an alternative approach of doing so. It is irrespective of genre, artists and albums so it asks the question of what features can connect a song to a mood.

We explore mood label extraction, dataset curation and classification to build an open-source pipeline for mood classification. To extract mood labels we explore mood theory in psychology which consist of "affect" models by Russell [1] and Thayer [2] and their corresponding adjectives that describe each mood category. We create datasets using these mood models and perform comparative analysis on them to identify the better mood model which would be representative of the class labels for the data. In order to create the datasets, we use open-source repositories – LastFM and MillionSong subset - with social media tags by listeners. The acquisition of lyrics for the creation of the lyrics datasets is done through LyricWikia API, an open-source API for extracting lyrics for a given song by an artist.

We classify the data using audio features and lyrics and explore existing techniques that have attempted to solve this task. An end-to-end open source classification pipeline is built using a CNN classifier and its performance is evaluated against existing methods. We create a simple and reproducible approach to music mood classification using natural language processing and deep learning. Natural language processing techniques such as using pre-trained word embeddings, are explored to prepare the data for classification.

Machine learning techniques with NLP approaches such as term frequency-inverse document frequency, continuous bag of words with Word2Vec [3] and distributed bag of words using Doc2Vec [4] models are explored. Deep Learning techniques via CNNs, Bi-directional LSTMs and CRNNs are applied to lyrics which are processed using a pre-trained word embeddings model. The CNN model achieves 71.0% classification accuracy, highest among the models developed. We establish the performance of our CNN model by validating against the approaches used by Bischoff et. al [5] on audio features with SVMs and Hu et. al [6] on lyrics with SVMs which achieved accuracies of 56.7% and 60.0% respectively.

EXISTING METHODS

Models using audio features in [5] and using lyrics in [6] are built on the valence-arousal scale of Thayer et. al [2] and achieve the highest accuracies of 56% and 60% respectively. Some approaches to mood-based recommendation such as in [7] have utilized a more recent model for mood distribution called the Geneva Emotional Music Scales [8], where the moods evoked by music alone are isolated and grouped together into labels. The labels were designed to be ones specifically pertaining to music. These labels were in turn used to create recommendation systems called MoodPlay [9].

Moodplay uses a similarity based recommendation by using the affect labels from the GEMS [3] scales. We test the labeling convention of all three models in this paper. Mood theory [1], [2] plays an important role in the labeling process. It is the backbone of the mood labeling process. For any data, the quality of the data set can be determined by how well the labels pertain to the data. The label extraction process in this paper is similar to previous work done in [5] where they collected mood labels with the help of All Music Guide at allmusic.com. The initial data included 6,000 songs and 178 mood labels. The labels were organized into (a) a 2-dimensional diagram using Thayer's model [2] and (b) MIREX mood clusters. This was then used to reduce the dataset to 1,192 songs. The audio features extracted from these songs included 240 audio features such as tempo, loudness, pitch classes, chroma etc. Then a (SVM) support vector machine was created to classify on the basis of audio features and a naïve bayes classifier was built on the basis of these tags. The technique yielded a significant improvement in the F-measure to 0.572 versus the reported accuracy of 56.4% with tag-only features and 43.2% for audio-only features.

A hybrid approach involved using lyrics and audio information by Hu et al. [6] using audio signals and lyrics for Pop and Rock songs. 9,000 audio recordings were obtained for this data. Lyrics were collected from Lyricwiki.org. Tags were then processed in order to create a vocabulary of mood words. The problem was a multi-class classification problem, decomposed into a collection of binary classification problems. The audio features used were spectral features and MFCC (Mel-Frequency Cepstral Coefficients) from MARSYAS [10]. SVMs were used on these features to obtain an initial accuracy of 60.0%. The 18-category binary classification problem achieved an accuracy of 61.7%. The drawback to this approach is that there is no generalized classification pipeline that can do the same for a new track. Since the problem is decomposed into binary classification tasks for each of the 18 genres, the song can be in any one of the classes that correspond to the classes that represent 'not genre'. We use a CNN for a multi-class classification approach different from the existing SVM based one versus all approach [6].

Hu et. al [6] designed a technique for sentence division of lyrics into mood groups to extract mood labels via a large collection of mood words. The first step was identifying a large set of affect words in Chinese. The collected words were mapped to into a 2-dimensional real space R-square. The second step involved mapping the words in the ANEW (a database for natural language processing) onto the R-square space. The lyrics collected were labeled into four quadrants of the Thayer [2] model. Using Manhattan distance to measure similarity between two sentences, they achieved an F-measure of 0.44. This approach is a great approach to try and recreate for a dataset for English music. Affect words are the foundation of mood classification tasks and mapping the words and calculating a distance measure to recognize the words closest to the affect words can be done via tf-idf based classification. With the NLP corpora being vast, lyrics can be tokenized and can be used to find the closest words using distance measures to

calculate word and document similarity. We investigate the two approaches for mood classification.

Deep learning approaches for text classification have been demonstrated to work well in recent years. Y. Kim [11] devised a CNN architecture that captures patterns in text through a vector matrix passed through a CNN as input. Each vector in the matrix represents a token or a word that is vectorized using word embeddings. This approach is now a universal approach with minor modifications and parameter tuning. The use of pre-trained word vectors that have been trained on vast dictionaries of words allow the pre-trained model capture more of the words and convert them into words embeddings. This step is vital as the quality of word embeddings is important for the learning process. A pre-trained model such as GloVe can capture a lot of words and create unique embeddings that can be used then to create the input embedding matrix. Even a shallow single layered CNN can learn quickly if there are more word-associations to learn. Complex models have trouble in learning word-label associations when the embeddings are not of a good quality. Using this approach in this paper, our approach is created. This approach allows the capturing of word combinations of greater lengths that typical n-gram approaches. The CNN's core is a lot of multiplications so it can quickly lean the correlation between occurrences of words and a bag of many words to a class label. The better approach among all the approaches in this paper to uses a CNN to classify moods as part of a mood classification pipeline.

In [12], K. Choi et. al demonstrated a CRNN architecture for music lyrics classification to pass the outputs of CNNs into an RNN to try and capture sequential information of lyrics. This paper explores both techniques to demonstrate the effectiveness of a CNN coupled with a pre-trained embedding model to be a faster and better model for capturing associations and patterns

in lyrics with respect to a mood class. This paper also adds to the work demonstrated in [14] to show that using CNNs for a multi-class mood classification task is the most viable approach at least for open source mood data as used in this paper. The CRNN used in this paper follows the work done in [12] and makes changes based on the nature of the data. The performance of the CRNN in this paper will be pretty good though not as good as the CNN with word embeddings. The runtime of a CRNN is significantly quicker than that of a Bi-Directional LSTM and it does try capturing patterns in the data that a CNN would theoretically miss. This however would still increase the time complexity of the pipeline. The CRNN performs well in situations where the associations between the words and class labels are complex and are based on contextual information that a RNN would typically capture.

BACKGROUND: EMOTION MODELS

We obtain mood data using labels based on the Circumplex Theory of Affect – Russell [1] and the affect model by Thayer [2]. These models distribute human emotions on a valence-arousal scale. In [1] and [2] they demonstrate adjective grouping to group terms under a single label. This helps reduce labels and categorize them into generalized groups.

TABLE I. RUSSEL'S EMOTION ADJECTIVES

Label	Emotions
Angry	alarmed, tense, angry, annoyed, afraid, distressed frustrated
Happy	aroused, astonished, excited, delighted, happy
Sad	miserable, sad, gloomy, depressed, bored, droopy, tired
Relaxed	sleepy, calm, relaxed, satisfied, content, at ease, serene, glad, pleased

TABLE II. THAYERS EMOTION ADJECTIVES

Label	Emotions
E2	Excited, happy, pleased
E2	Annoying, angry, nervous
E2	Sad, bored, sleepy
E2	Relaxed, peaceful, calm

In 2008, M. Zentner [8] explored emotions that would correspond specifically to music and created the Geneva Emotional Music Scales. This research was done as an attempt to establish 9 mood clusters that are exclusively evoked by music and also used the adjective group to address the challenges that arise while labeling emotions. These mood clusters each consist of a dictionary of music-relevant emotion terms that correspond to each of the 9 emotions. The

studies conducted in this paper established that there was a significant difference in the frequency ratings of musical emotions versus everyday emotions. The dictionaries can be used to segregate extracted music tags for each song and assign them to an emotion cluster. Additional audio analysis can then be performed in order to understand the different audio features that correlate to a song's mood/emotion.

TABLE III. GEMS EMOTION ADJECTIVES

Label	Emotions
Wonder	happy, filled with wonder, allured, dazzled, moved, admiring
Transcendence	inspired, feeling of transcendence, feeling of spirituality, thrills, fascinated, overwhelmed
Tenderness	in love, sensual, affectionate, tender, mellowed
Nostalgia	sentimental, dreamy, nostalgic, melancholic
Peacefulness	calm, relaxed, serene, soothed, meditative
Power	energetic, triumphant, fiery, strong, heroic
Joyful Activation	stimulated, joyful, animated, feel like dancing, amused, bouncy
Tension	agitated, nervous, tense, impatient, irritated
Sadness	sad, sorrowful, tearful

DATASET

The Million Song Dataset [13] is chosen for this task. It consists of audio features as well as a vast database of tags from LastFM. The LastFM database consists of corresponding track IDs that can be used to create a tagged dataset pertaining to a mood model. As the dataset is open-source, it is ideal for the creating a mood dataset to be used in future work. The dataset makes available the following features: Tempo, Time Signature, Key, Mode, Duration, Loudness, Timbre (across 12 segments). Tempo is the measure of the number of beats in a minute that measures the pace of a song. Time signature explains the structure of a song in terms of the number of beats per bar. The key of a song determines the combination of sharps and/or flats of notes that are present in the song. Duration is the length in seconds of the song. Mode is the modality of the song. Loudness is the measure of decibels in a song. Timbre is used to represent the quality and texture of a song and is measured numerically using MFCCs. We then retrieve the lyrics of each track using the scraper that we built using the LyricWikia API. The scraper retrieves the lyrics and appends it to the dataset via a data frame and saves it as a '.csv' file.

Tag Extraction

The Million Song Dataset [13] is linked to the LastFM database. Each track has a list of tags that can be searched and extracted. The database is available as an SQLite Database. In order to extract the tags, we create three models for each set of mood classes – Russell [1], Thayer [2], and GEMS [3]. An SQLite scraper is written that extracts track IDs which contain a mood as a tag. The list of moods in each class are as follows.

Russell:

Angry = [alarmed, tense, angry, annoyed, afraid, distressed frustrated]

Happy = [aroused, astonished, excited, delighted, happy]

Sad = [miserable, sad, gloomy, depressed, bored, droopy, tired]

Relaxed = [sleepy, calm, relaxed, satisfied, content, at ease, serene, glad, pleased]

Russell's circumplex model [1] is the first established and most widely used model from mood theory to classify moods. The model provides a spatial representation of affective concepts on a valence-arousal scale. Though there have been attempts to use Thayer's model, Russell's model has proven to be more effective so far. This analysis enabled building our datasets with each mood model to determine which model is a best fit for this paper's data.

Thayer:

E1 = [excited, happy, pleased]

E2 = [annoying, angry, nervous]

E3 = [sad, bored, sleepy]

E4 = [relaxed, peaceful, calm]

Thayer [2] also uses a 2-D valence-arousal model. It clusters moods into four quadrants. Each denotes one type of mood. Horizontal dimension being pleasantness and the vertical dimension indicating the level of energy, it creates a model similar to Russell's though with lesser moods. This model has worked out in binary mood model classification by generalizing moods into two categories: Happy and Sad.

GEMS (Geneva Emotional Music Scales):

Wonder = [happy, filled with wonder, allured, dazzled, moved, admiring]

Transcendence = [inspired, feeling of transcendence, feeling of spirituality, thrills, fascinated, overwhelmed]

Tenderness = [in love, sensual, affectionate, tender, mellowed]

Nostalgia = [sentimental, dreamy, nostalgic, melancholic]

Peacefulness = [calm, relaxed, serene, soothed, meditative]

Power = [energetic, triumphant, fiery, strong, heroic]

Joyful Activation = [stimulated, joyful, animated, feel like dancing, amused, bouncy]

Tension = [agitated, nervous, tense, impatient, irritated]

Sadness = [sad, sorrowful, tearful]

GEMS [8] are a set of mood classes created to isolate moods that pertain solely to music, resulting from a survey from the University of Geneva. It creates a music-affect mood model in 2008. This model consists of more adjectives and mood classes in total and are also based on the arousal valence scale. This is the most recent mood model for music mood analysis. The adjectives from this dataset did not retrieve enough data points for a classification task.

Dataset Creation:

The mood tags, once queried through the LastFM SQLite database, produce a labeled list of tracks. The list of track IDs are then queried through the Million Song Database to extract the tracks and corresponding audio features as shown in Fig. 1.

The track and artist names are used as input queries through the LyricWikia API to extract the song lyrics. Finally, three datasets are created for each mood model. This data helps establish the ground truth for the classification pipeline.

Comparative studies are performed across the datasets to decide which ones have the most accurate labels for mood classification.

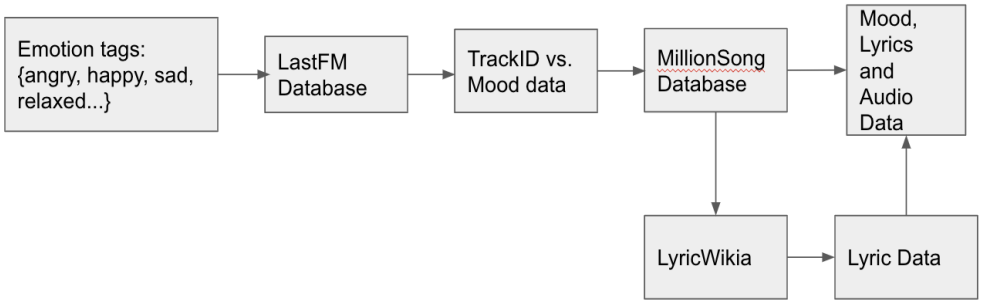

Figure 1 Label extraction and dataset creation

DATA PREPROCESSING

Data preprocessing for lyrical data involves the following steps depending on the task. The tasks involved in our natural language models are – tf-idf (term frequency-inverse document frequency) classification, word2vec [3] classification, doc2vec [4] classification and pre-trained word embeddings for classification using deep learning. Our preprocessing tasks are: Special Character Removal, Lowercasing, Tokenization, Lemmatization, Stop-word removal. Special character removal is the process of cleaning text data by removing non-textual characters that can interfere with the information retrieval process through language. Lowercasing of text allows the text information to be uniform in format, allowing for better precision in creating word embeddings. It also allows language parsing libraries such as 'nltk', 'spacy' and 'genism' to be able to pick up most of the words in the data. Tokenization of data converts the data into tokens, i.e., singular words that become data points or the building blocks of data points. Tokenization is sometimes followed by lemmatization. Lemmatization – a process similar to stemming, except that it ensures that the root form of the word is a lexically correct term with independent meaning. We perform lemmatization in word2vec vectorization models and in models that calculate tf-idf scores to connects the scores to class labels.

However, lemmatization is not a guaranteed way to improve model performance and must be compared with models without lemmatization to determine its requirement. Stop-word removal removes filler words such as 'the,' 'a,' 'umm,' etc. It is a technique that reduces dimensionality and removes noise from the data. For this data, we use special character removal, lowercasing, and tokenization across all models. We use lemmatization and stop-word removal in tf-idf classification and word2vec-based classification variants of classification models.

EXPERIMENTS AND RESULTS

Mood Classification using Audio:

Previous methods dealing with audio feature based music classification predominantly use MFCCs to understand timbre [10], [14], [15]. The Million Song Dataset [5] consists of the timbre segments along with the other audio features. Tempo is the measure of the number of beats in a minute that measures the pace of a song. Time signature explains the structure of a song in terms of the number of beats per bar. The key of a song determines the combination of sharps and/or flats of notes that are present in the song. Mode is the modality of the song. Loudness is the measure of decibels in a song. Timbre is used to represent the quality and texture of a song and is measured numerically using MFCC.

To compare to the approach in [5], classification is performed on these datasets, and the results are compared. The classification is done using 5-fold cross validation to avoid any errors and possible overfitting. The highest accuracy yielded in these models is 57.93% using a logistic regression classifier on the audio features with labels from Russell's [1] model. Thayer's [2] model performs similarly but the labels yield slightly lower classification accuracy of 56.94%. Due to the overlap of labels between Russell's and Thayer's models, a comparative classification experiment is implemented and from the results we observe better accuracies with Russell's dataset. The GEMS dataset is eliminated, as its labels did not yield enough data points and consisted of overlapping samples across class labels.

TABLE IV. MOOD CLASSIFICATION USING AUDIO FEATURES

Dataset	Classifiers					
	SVM	***Rand. Forest***	***KNN***	***MLP***	***L. Reg***	***N. Bayes***
MSD+ Thayer	36.64, (0.58)	53.43, (1.10)	47.5, (0.89)	39.10, (0.03)	56.94,(1 .89)	45.57, (2.8)
MSD+ Russell	37.20, (.84)	53.5, (1.51)	47.41, (1.48)	39.3, (0.04)	57.93, (1.07)	44.06, (2.26)

Mood Classification using Lyrics with Traditional Machine Learning Classifiers:

Lyric-based multi-mood classification models [16], [17], [18], [19] have used the bag-of-words, manhattan distance-based and the tf-idf methods. SVMs, naïve bayes and random forest classifiers have been used in previous attempts [6] at mood classification. We classify the data using these classifiers along with additional classifiers – logistic regression, multi-layer perceptron. The tf-idf values give a maximum accuracy of 66.85% using logistic regression. The model performs best without lemmatization but with the inclusion of a stop-word removal process in data preparation. To try and improve the model, a vector representation of the data is created using Google News Vectors. This model gives a highest accuracy of 60.11% using Random Forest Classifier. All classification is done with 5-fold cross validation. The results prove that Russell's model is the more effective labeling model for mood.

TABLE V. MOOD CLASSIFICATION USING EXISTING METHODS ON LYRICS

Dataset	Classifiers			
	SVM	***Rand. Forest***	***L. Reg***	***N. Bayes***
MSD+ Thayer Word2vec	38.15, (0.07)	53.24, (2.68)	52.33, (2.69)	42.03, (2.70)
MSD+ Russell Word2vec	44.14, (0.10)	60.11, (1.79)	54.52, (2.07)	43.23, (3.42)
MSD+ Thayer Doc2vec	54.72, (2.19)	52.80, (2.42)	47.23, (0.75)	50.58, (1.23)
MSD+ Russell Doc2vec	61.14, (0.28)	58.34, (2.36)	53.00, (1.24)	53.60, (3.29)
MSD+ Thayer Tf-Idf	39.01, (0.02)	56.82, (0.50)	55.85, (1.27)	57.39, (0.88)
MSD+ Russell Tf-Idf	44.75, (.045	66.67, (1.83)	66.85, (2.09)	55.07, (1.53)

We observe that lyric data performance is higher – 66.85% in comparison to audio features performance and becomes the baseline for classification with deep learning. We use GloVe [20] (Global Vectors for Word Embeddings) for creating word embeddings for the lyrics. This is the input to the lyrics-based deep learning model.

GloVe is an unsupervised learning algorithm to generate word vectors. It is a good choice for creating word embedding matrices for neural networks for the task of text classification. The GloVe 6B tokens model with 400K vocabulary is a good vectorizer with fast loading times for text and image data. GloVe uses a count-based model that is advantageous when used with neural networks as it is count based and can capture more combinations than an n-gram based approach. We use this to create an embedding matrix as input for deep learning models.

To create the matrix, we reduce sequences longer than 1000 and use padding for shorter word vectors. The word embeddings matrix is then split into training and validation data – 80% training and 20% validation. Experiments are conducted using CNNs (Convolutional Neural

Networks), Bi-directional LSTM (Long-Short-Term-Memory) Networks and CRNNs (Convolutional-Recurrent Neural Networks).

CNNs for text classification:

CNNs have been effective in sentence classification as demonstrated by Y. Kim [6]. The CNN architecture, that we use consists of three convolution layers of kernel sizes (2,5,10), which allows for the detection of patterns of multiple sizes. Patterns can be expressions or word n-grams such as 'I like', 'pretty great,' and therefore the CNN can identify them in the sentence regardless of their position. The model takes in a matrix of word embeddings as input where each row of the matrix corresponds to a token. Since we used GloVe, the token is a word vector. The filters slide over full rows of the matrix (words). The working of a CNN for natural language processing is through a bag of words model. The CNN filters when used with GloVe vectors, allow the model to capture more combinations and patterns of word occurrences. CNNs perfom a lot of multiplication to classify the data. They can run on CPUs without needing expensive hardware.

CNNs are efficient in terms of representation compared to an n-gram model. Anything more than 3-grams can become expensive with a large vocabulary. This allows for the capture of more features over approaches that use the n-gram or tf-idf approach with an SVM or logistic regression classifier. The CNN uses three concatenated convolutional layer with max-pooling, two dense layers with 'relu' activation, a dropout layer and the output is passed through a 'softmax' activation function. 'Categorical crossentropy' is used as a loss function to classify the data into multiple classes: angry, happy, sad, relaxed.

The model yielded an accuracy of 71% over 200 steps.

The classification report is as follows:

TABLE VI. CLASSIFICATION REPORT

Class	Precision	Recall	F1-Score	Support
0	1.00	0.18	0.31	11
1	0.72	0.90	0.80	95
2	0.77	0.43	0.55	95
3	0.70	0.70	0.70	166

The confusion matrix is as follows:

[2 4 0 5]

[0 179 4 15]

[0 25 41 29]

[0 42 8 116]

Labels: Angry: 0, Happy: 1, Relaxed: 2, Sad: 3

From the confusion matrix, we observe that the model has difficulty learning the features for class 0: Angry, this is possibly due to fewer data points to learn features from. We could only extract 73 songs for this label. The 2 labels that the model learned the best are Happy and Sad as the data points were the highest. Angry and Relaxed did not yield as many data points.

Especially Angry as it only had 73 data points. The sampling from these data points was not inad equate as per the sample size requirement for the respective population sizes from the analysis sh own below:

Type 1 error rates: [0.0 0.26 0.03 0.16]

Type 2 error rates: [0.82 0.1 0.57 0.3]

Population: {0: 73, 1: 1053, 2: 437, 3: 790}

Sample: {0: 62, 1: 855, 2: 342, 3: 624}

For class 0, according to a 95% CI, the sample size needed is 226, we used 62,
the confidence level of the observed predicted accuracy is 84.73% (one-tailed)

For class 1, according to a 95% CI, the sample size needed for is 138, we used – 855

For class 2, according to a 95% CI, the sample size needed is 376, we used – 342
The confidence level of the observed predicted accuracy is 96.91% (one-tailed)

For class 3, according to a 95% CI, the sample size needed is 322, we used – 624

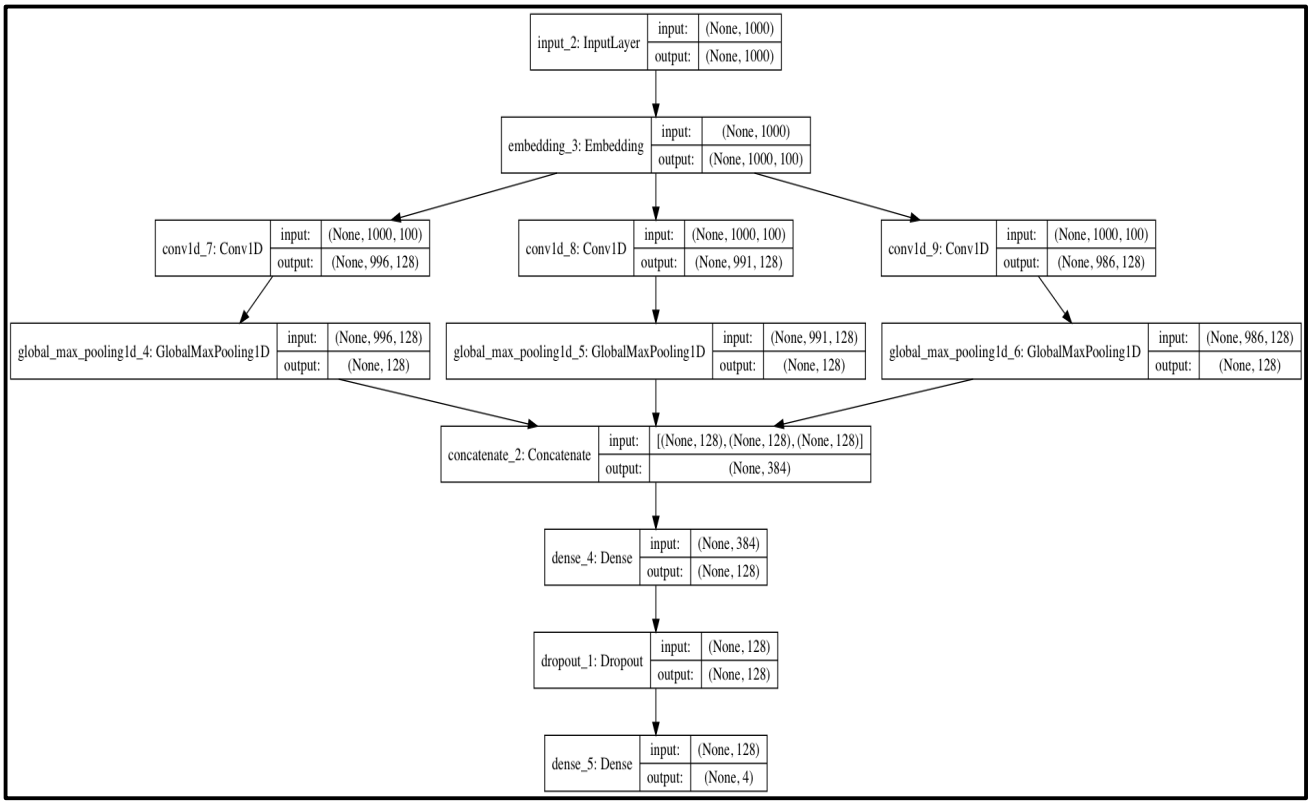

Figure 2 CNN architecture

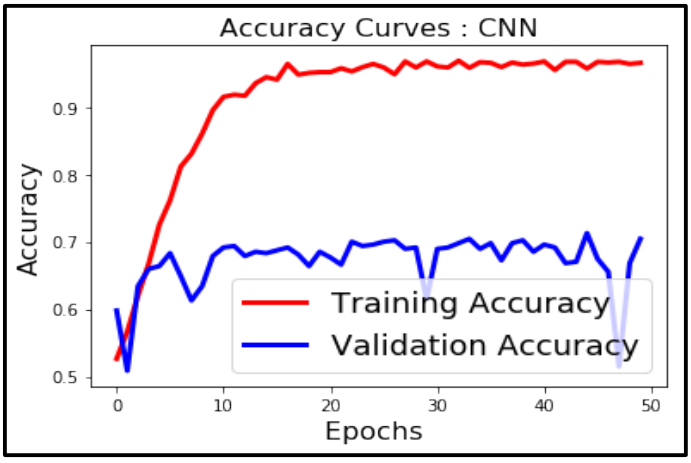

Figure 3 CNN performance

Bi-directional LSTM for text classification:

A bi-directional LSTM is used to capture sequential information. Pre-trained word embeddings can help capture the semantic representation of texts. We implement a bi-directional LSTM model for our data. It yields an accuracy of 69%. The classifier is trained over 50 steps on a CPU. The architecture of the bi-directional LSTM uses hidden layer size of 100 and use a 'softmax' activation with loss - 'categorical crossentropy'. The training times are long for this classifier without a GPU.

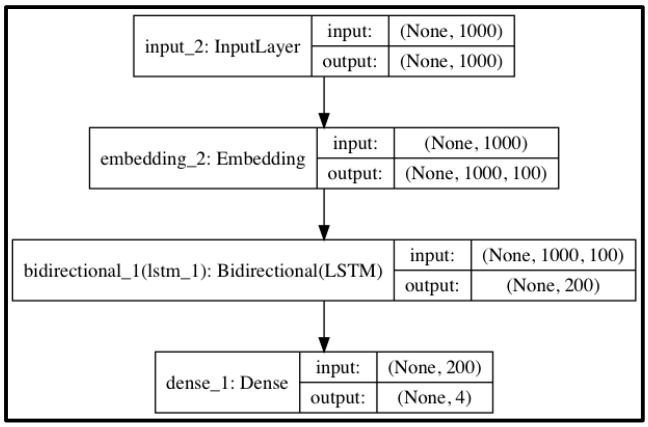

Figure 4 Bi-Directional LSTM architecture

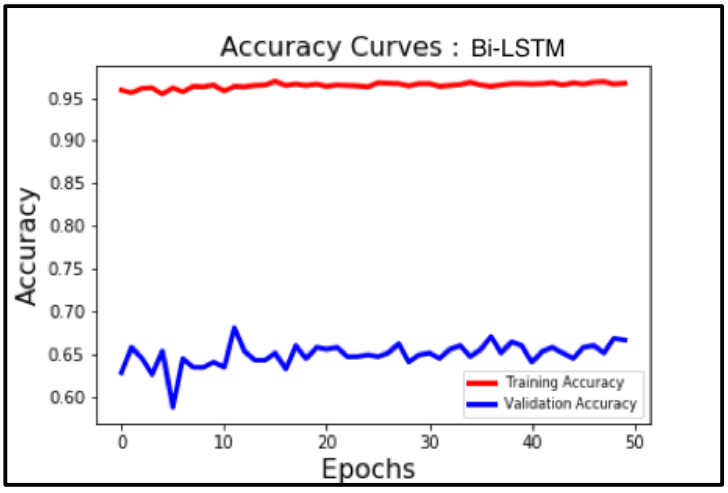

Figure 5 Bi-Directional LSTM performance

CRNN (CNN with LSTM) for text classification:

Another model that was tested was a CRNN which used 3 convolution layers that enter an LSTM with hidden layer size: 64. CRNNs have been used for music classification and have shown to perform well with lyric-data [16]. We use the 'categorical crossentropy' for loss and a 'softmax' activation function. The model gives an accuracy of 67.04% across 100 steps and takes relatively less time to train than an RNN but still trains slower than a CNN.

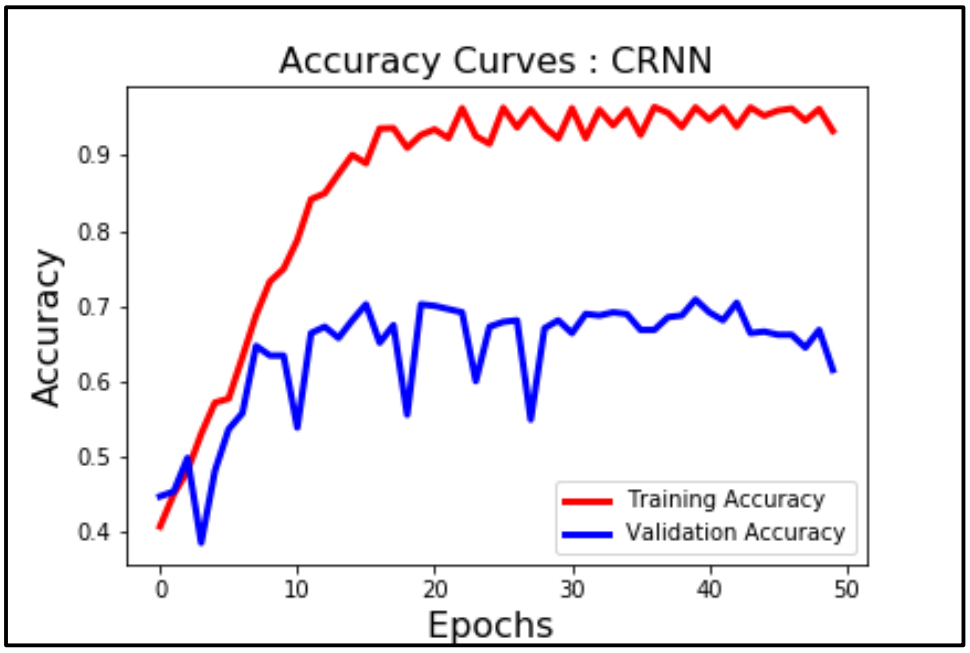

Figure 6 CRNN performance

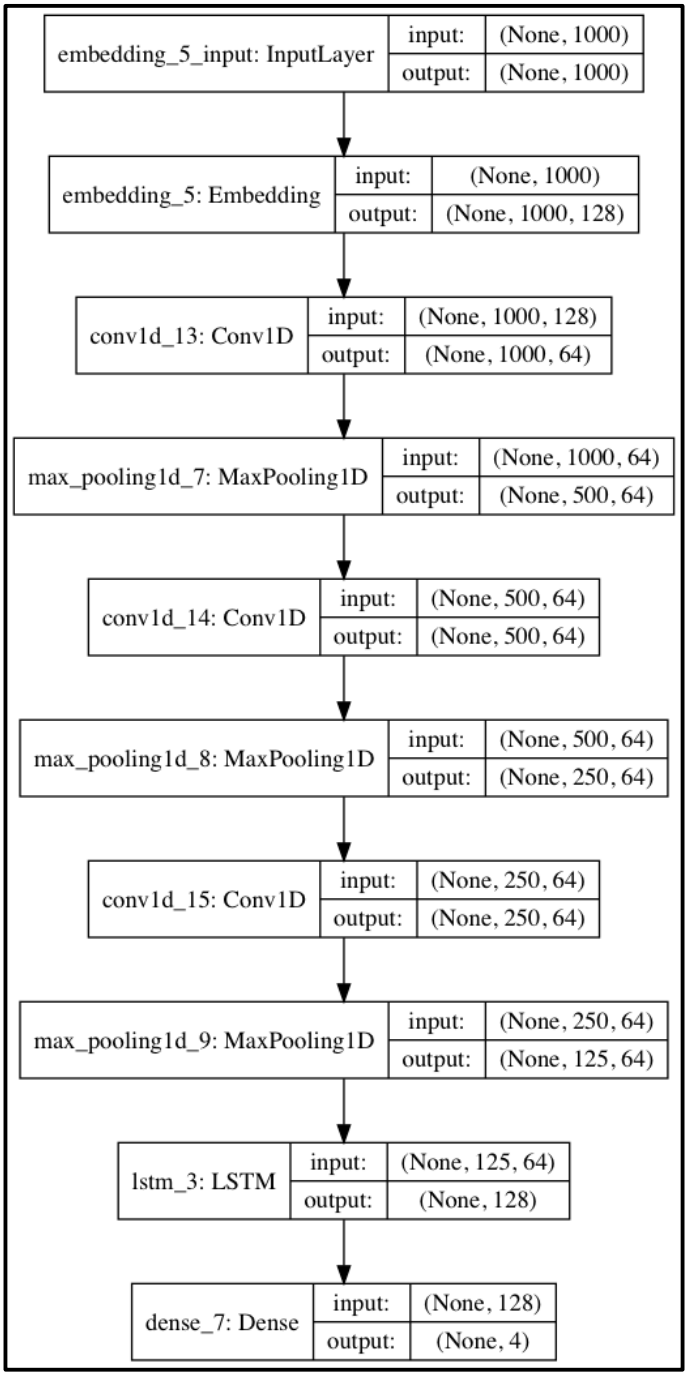

Figure 7 CRNN architecture

EVALUATION

The models perform better than the models that use traditional machine learning classifiers with NLP techniques. We see the highest results from a CNN which also trains quickly. The RNN performance is close to that of a CNN but it takes a long time to converge and therefore is not the most reliable classification model. The CRNN was an attempt to bridge the gap between the two models to capture more patterns as well as reduce training times but still performs slower than the CNN model. The results of the three models are tabulated in Table VII.

TABLE VII. ACCURACY CHART

Classifier	***Accuracy(%)***
CNN	71.00
Bi-LSTM	69.01
CRNN	67.04
TF-IDF	66.85
Word2Vec	60.11
Doc2Vec	61.14

CONCLUSION

The experiments conducted in this paper explore the techniques that are try to solve a mood classification problem for music data. The results show that lyrics provide excellent linguistic data that tie music to mood labels and provide an appropriate grouping of songs. The paper develops an efficient classification pipeline that can be re-created using open source data and can be used to classify additional data points with 71% accuracy over existing approaches by [6] that achieved 60.0% accuracy using SVMs on lyrics-data. It also demonstrates that CNNs are a good solution to English-language based text classification problems and are quick and easy to train without requiring expensive computational hardware. Our experiments conclude that lyrics classification of music using emotion labels from Russell's [1] circumplex model via a CNN can establish a strong open source approach to music mood classification and can successfully classify data better than methods that use a count-based approach. GloVe [20] embeddings are a powerful source of word embeddings for words in the English language and is a great pre-trained model for this task. Mood classification of music can thus be successfully done through the approach provided in this paper with maximum results and through open source approaches with the help of lyrics and natural language processing using a three-layer, shallow Convolutional Neural Network without high computational requirements.

REFERENCES

[1] J. A. Russell, “A circumplex model of affect.,” Journal of Personality and Social Psychology, vol. 39, no. 6, pp. 1161–1178, 1980.

[2] Thayer, Robert E. “Mood.” Encyclopedia of Psychology, Vol. 5., pp. 294–295., doi:10.1037/10520-125.

[3] T. Mikolov. "Distributed Representation of Words and Phrases" [online] Papers.nips.cc. Available at: https://papers.nips.cc/paper/5021-distributed-representations-of-words-and-phrases-and-their-compositionality.pdf [Accessed 22 May 2019].

[4] Q. Lee and T. Mikolov "Distributed Representations of Sentences and Documents" [online] Cs.stanford.edu. Available at: https://cs.stanford.edu/~quocle/paragraph_vector.pdf [Accessed 22 May 2019].

[5] K. Bischoff, C. Firan, R. Paiu, W. Nejdl, C. Laurier and M. Sordo, "Music Mood and Theme Classification a Hybrid Approach", Mtg.upf.edu, 2018. [online]. Available: http://mtg.upf.edu/node/1465. [Accessed: 10- Dec- 2018].

[6] X. Hu, J. Stephen, D. Andreas, & F. Ehmann, “Lyric text mining in music mood classification.,” in Proc of the Int. Society for Music Info. Retrieval Conference, 2009, pp. 2-209.

[8] M. Zentner, D. Grandjean, and K. R. Scherer, “Emotions evoked by the sound of music: Characterization, classification, and measurement.,” Emotion, vol. 8, no. 4, pp. 494–521, 2008.

[9] I. Andjelkovic, D. Parra, and J. O’Donovan, “Moodplay: Interactive music recommendation based on Artists’ mood similarity,” Int. Journ. of Human-Computer Studies, vol. 121, pp. 142–159, 2019.

[10] G. Tzanetakis, “Music analysis, retrieval and synthesis of audio signals MARSYAS,” Proc. of the 17th ACM int. conf. on Multimedia - MM 09, 2009.

[11] Y. Kim, “Convolutional Neural Networks for Sentence Classification,” Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), 2014.

[12] K. Choi, G. Fazekas, M. Sandler, and K. Cho, “Convolutional recurrent neural networks for music classification,” 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2017.

[13] T. Bertin-Mahieux, D. Ellis, and P. Lamere, “The Million Song Dataset,” Proceedings of the 12th Intl. Soc. for Music Information Retrieval Conf., 2011.

[14] B. K. Baniya, C. S. Hong, and J. Lee, “Nearest multi-prototype based music mood classification,” 2015 IEEE/ACIS 14th International Conference on Computer and Information Science (ICIS), 2015.

[15] H. Shahmansouri and J. Z. Zhang, “An empirical study on mood classification in music through computational approaches,” 2016 3rd International Conference on Systems and Informatics (ICSAI), 2016.

[16] Y. An, S. Sun, and S. Wang, “Naive Bayes classifiers for music emotion classification based on lyrics,” 2017 IEEE/ACIS 16th Int. Conf. on Computer and Information Science (ICIS), 2017.

[17] F. H. Rachman, R. Sarno, and C. Fatichah, “Music Emotion Classification based on Lyrics-Audio using Corpus based Emotion,” Int. Journ. of Electrical and Computer Engineering (IJECE), vol. 8, no. 3, p. 1720, Jan. 2018.

[18] C. Laurier, J. Grivolla, and P. Herrera, “Multimodal Music Mood Classification Using Audio and Lyrics,” 2008 Seventh International Conference on Machine Learning and Applications, 2008.

[19] Y. Hu, X. Chen, and D. Yang, “Lyric-based Song Emotion Detection with Affective Lexicon and Fuzzy Clustering Method.,” in ISMIR, 2009, pp. 123–128.

[20] J. Pennington, “GloVe: Global Vectors for Word Representation.,” [online] Nlp.stanford.edu. Available at: https://nlp.stanford.edu/projects/glove/ [Accessed 22 May 2019].