Mood Classification Using Song Lyrics
Introduction to Natural Language Processing - Project

Abhiram Ravi Bharadwaj
Rochester Institute of Technology
Department of Computer Science
ab8136@rit.edu

Abstract

With the recent advancements in music streaming services like Spotify, YouTube Music, Amazon Prime Music etc, it has become a major avenue to suggest music to the audience. Suggesting valid music to the audience requires understanding the mood of the audience and recommending songs based on the mood. An obvious necessity is to be able to classify music into the various different moods so as to be able to recommend music. This project aims at successfully classifying musing into one of various different moods.

1 Introduction

The current technology allows for people to be connected to the internet at all times. This means that there is very little need for any media to be stored locally on any device. That media includes music as well. With the recent advancements in music streaming services like Spotify, YouTube Music, Amazon Prime Music etc, it has become a major avenue to suggest music to the audience. Suggesting valid music to the audience requires understanding the mood of the audience and recommending songs based on the mood. An obvious necessity is to be able to classify music into the various different moods so as to be able to recommend music.
This project aims at successfully classifying musing into one of various different moods. This task can be accomplished in two ways, either by considering the audio of the songs or by looking at the lyrics in the songs. This project looks at the latter approach by classifying lyrics into various moods. There are some particular steps that would need to be followed to accomplish this:

- **Data collection** is the first step in any machine learning process. Same is the case in this project. A corpus of song lyrics was needed. A corpus of nearly 2,000 song lyrics from four genres:

  1. Rock
  2. Country
  3. Rap/Hip Hop
  4. Reggae

  was obtained.

- **Data annotation** is the process of labelling data into the various classes of interest. In this project the data was annotated using the GEMS scale for music emotion. At least one category was assinged to each instance of the data. The categories and definitions are mentioned in Table 1.

Table 1: List of categories with the definition of the categories according to GEMS.

Category	Definition
Amazement	Feeling of wonder and happiness.
Solemnity	Feeling of transcendence.
Tenderness	Sensuality, affection, feeling of love.
Nostalgia	Dreamy, melancholy, sentimental feelings.
Calmness	Relaxation, serenity, meditativeness.
Power	Feeling strong, heroic, triumphant, energetic.
Joyful Activation	Feels like dancing, bouncy feeling, animated, amused.
Tension	Nervous, impatient, irritated.
Sadness	Depressed, sorrowful.

- **Data cleaning** is essentially preprocessing data. Once the data has been annotated, it is prone to having a lot of junk values as well. The quality of work of annotators cannot ever be guaranteed since annotation is subjective. So instead of annotating the data again, it

would just be easier and save a lot of resources to just clean the data. This is the step where junk data would also be removed.
Most song lyrics are usually obtained by speech to text machines (Gupta et al., 2019; Kruspe, 2016; Mesaros and Virtanen, 2010) and this usually leads to retrieval of some unwanted data as well. This could be due to the singer humming a tune or instruments sounding like people etc.

- **Classification** is the logical next step once the data has been cleaned and ready to be consumed by the machine learing models. Classification involves training a model into being able to classify the song lyrics into one of the classes provided. Typically multiple classification algorithms like k-nearest neighbour (Guo et al., 2004), support vector networks (Cortes and Vapnik, 1995). In this project we look at at least two different classification algorithms and compare their results. This process is discussed in more detail in section 4.

- **Analysis**. Once the classification model is built, it needs to be tested to check how well it performs. This could be done by looking at various scores and also visualisations. In this project, visualisations are not considered.

Section 2 briefly talks about some related work to this project. In section 3 the data used in this project is explained with some of the preprocessing done on it. The crux of the work done is explained in section 4. Section 5 explains the results obtained from the experiments and section 6 concludes this project's outcomes.

2 Related Work

Describe related work on lyrics/music classification. 1pg Attributing to the increased importance of recommender systems, a lot of recent research work has focussed on building better recommendation systems. Music also has grown into needing a recommendation system due to the vast popularity of streaming services. This need draws a lot of attention and some good research has been done. One such work is by Giammusso et al. (Giammusso et al.) who propose an approach to classifying songs into four different classes:

- relaxed
- happy
- sad
- angry

In their work, they look at identifying stylometric, structural, orientation and vocabulary based features which they extract from lyrics. These featues include attributes like the proportion of verbs in various tenses and also some POS tags.
Recommendation systems also rely on identifying what songs are popular or rather in predicting what songs might get popular and suggesting those songs to its subscribers. This area was tapped on by Barman et al. (Barman et al., 2019). Of the various different approaches to identifying popular songs, they use lyrics. They note that the style of lyrics have a very obvious difference between popular songs and other songs. Another aspect that they consider is the bias in songs. This, according to them is similar to the bias that humans show. This tells that song lyrics reflect the biases in society. Truly the work of Giammusso et al.(Giammusso et al.) and Barman et al. (Barman et al., 2019) can be thought of as previous work rather than related work.
The research that is closest to what this project aims at doing was done by Hu et al. (Hu et al., 2009). They showcase how important it is in evaluating music information retrieval systems that can access the mood dimension of music in an automated manner. Their work looks deeply into twenty two systems that have been evaluated and looks to address the important issues that were raised during those evaluations. They note that due to the subjectivity of music perception, arriving at a ground truth for the mood of a song can be difficult. This is one of the issues that was encountered during the course of this project as well. Asking people to annotate data is easy but there is no strong distinction between what is right and what is wrong. This project can be construed as a miniature of the work done by Hu et al. (Hu et al., 2009). Although Hu et al. (Hu et al., 2009) used an SVM with a linear kernel and default parameters, this project will test other models and fine tune the parameters aiming to improve the accuracy of classification.
A problem that is similar to mood classification is genre detection. Song prediction can be done hierarchically as to first predict the mood and then predict the song. Selecting the song could be done by

choosing a genre so it can be seen that genre and mood are corelated. Fell et al. (Fell and Sporleder, 2014) build a lyrics based classification model that performs genre detection, distingushing best and worst songs and determining the approximate publication time of a song. Looking at all these research, it is evident that performing bag of words as described in section 4.2.1 is an important task.

3 Data

Brief description of the dataset. How many instances are in each class. How many tokens. Where does the dataset come from. How was it annotated. 0.5pg
As mention in section 1, the data consisted of around 2,000 instances. Once the data was collected, a group of 24 students from Rochester Institute of Technology were assigned the task of annotating the data. Each one of the 24 were requested to annotate a minimum of 200 instances. A particular instance was annotated by three different students to ensure the best possible annotation. Students were instructed to assign every record at least one class among the given nine class labels.
Once the annotation was done, every record had more than one class label associated with it. As part of the data pre-processing, the data was converted into two forms:

1. The first type was the **single label**. In this dataset, all the records were assigned just one class. The class assigned was decided by a max-pooling from all the annotations by the students. A sample of this data is shown in Figure 1.

	artist	genre	title	album	year	lyrics	label
0	Nirvana	Rock	You Know You're Right	Nirvana	2002.0	I will never bother you\nI will never promise ...	Sadness
1	Damian Marley	Reggae	Here We Go	Stony Hill	2017.0	Here we go\nMy big ego is gonna get me in trou...	Tension
2	The Mission UK	Rock	Jade	Another Fall from Grace	2016.0	She came as Lolita dressed as Venus\nAnd adorn...	Tenderness
3	UB40	Reggae	Food For Thought	Signing Off	1980.0	Ivory Madonna, dying in the dust\nWaiting for ...	Sadness
4	Johnny Cash	Country	I've Been Everywhere	American II: Unchained	1996.0	I was totin' my pack along the dusty Winnemucc...	Sadness

Figure 1: Sample of the single labelled data.

2. The second type was the **multi label**. In this dataset, all of the unique class labels that the students had assigned were present. A sample of this data is shown in Figure 2.

	artist	genre	title	album	year	lyrics	labels
0	Nirvana	Rock	You Know You're Right	Nirvana	2002.0	I will never bother you\nI will never promise ...	Calmness, Sadness
1	Damian Marley	Reggae	Here We Go	Stony Hill	2017.0	Here we go\nMy big ego is gonna get me in trou...	Power, Tension
2	The Mission UK	Rock	Jade	Another Fall from Grace	2016.0	She came as Lolita dressed as Venus\nAnd adorn...	Amazement, Calmness, Solemnity, Tenderness
3	UB40	Reggae	Food For Thought	Signing Off	1980.0	Ivory Madonna, dying in the dust\nWaiting for ...	Joyful activation, Sadness, Tension
4	Johnny Cash	Country	I've Been Everywhere	American II: Unchained	1996.0	I was totin' my pack along the dusty Winnemucc...	Amazement, Calmness, Joyful activation

Figure 2: Sample of the multi labelled data.

For the purpose of this project, only the single labelled data is used. Specifically, the data for three classes is extracted and a model is built to classify songs into one of these three classes. The classes selected are:

- Sadness
- Tenderness
- Tension

Table 2 shows the distribution of data into these three classes. These three classes had the most number of instances in them and hence these were chosen.

Table 2: Distribution into the three classes in single labelled data.

Category	Count
Sadness	569
Tenderness	326
Tension	265

4 Methods

After the data has been curated, it is now ready to be consumed by machine learning models.

4.1 Technology Used

A bunch of technologies and libraries were used in building this song classifier. Firstly, the data was annotated using LightTag (lig). LightTag aids in collaboration of annotators. It does not restrict one annotator from working while other might be too. LightTag also provides a simplistic and holistic design so that annotating becomes hassle free.
For the main implementation of the models, Jupyter Notebook (jup) are used. Jupyter Notebook helps increase the productivity by saving previously run code. It allows for a way to run certain

segments of the code and store the results in memory. That way every block of code can be individually built and shipped. In pipelines like machine learning where certain operations are very heavy, for instance loading huge amounts of data, this could be useful since data can be loaded once but varoius different operations can be performed on it.
A bunch of packages and frameworks were used to build the models. The data was loaded and stored in pandas (pan) dataframes. Most of the machine learning models were built using libraries from scikit-learn (skl).

4.2 Models

Every machine learning model involves various parameters. Building a good model involves tuning the model by changing these parameters and achieving a good model. Deciding what model is good involves comparing them with various metrics. Models are first trained on data but to check how good the model is, it needs to be tested on some data as well. To do so, the given dataset is split into training and testing data.
Every model learns by predicting a class for data points and then comparing it with the ground truth which is the annotated labels. Once the model is fit on the training data, it is evaluated on the testing data.
Humans understand language but that is not the case with computers. Computers need numbers to crunch. Thus it is required to convert the given lyrics into numbers that represent the lyrics. This cannot be as trivial as representing a string of ASCII values of all the characters in the string. Hence a technique called term frequency-inverse document frequency(tf-idf) is computed. tf-idf reflects the importance of words in a given corpus. In this project, the number of features is capped at 3,000.

4.2.1 Bag of Words

Bag of words is a model typically used in natural language processing where every document is represented as a bad of its words without regarding word order or grammar but retaining only multiplicity. In this project, unigrams, bigrams and trigrams are considered in all possible pairs. Each one of the model described below is run on all the different bag of words to check which would yield the best result.

4.2.2 Supply Vector Classifier

The very first model that was considered is the support vector classifier (SVC). SVC is a binary classifier and cannot be natively used as a multi class classifier. A SVC fits the data provided by computing a hyperplane that divides the two classes of data. To use a SVC as a multi class classifier, some tweaks would be needed to be made. A hierarchy of classification can be performed where each class is selected and classification is performed in a binary fashion as to whether a data point belongs to the class or not. Then among the data classified as not belonging to the class, another class is chosen and the same is repeated. This way in each iteration data points from one class is isolated. There are various different functions that can be used as kernels morphing the SVC into different forms. In this project, two such functions were used:

- **Linear -** In this case, a linear kernel is used to determine the hyperplane.
- **RBF -** In this case, a non-linear hyperplane is used.

Apart from the two mentioned, a polynomial kernel can be used too with varying degrees. In this project only the linear and rbf kernels were used in order to save time. SVC in general are known to be slow and rbf is known to perform the best in nlp. The work of Hu et al. (Hu et al., 2009) also confirms that polynomial kernels did not yield better results. This project uses the Linear SVC as the baseline model.

4.2.3 K-Nearest Neighbors

The next model to be considered was the k-nearest neighbors. This model starts off by considering k random centroids and during training constantly updates the centroids so that they are as close to the different class centres as possible. For every instance in the new (test) dataset, the classes of its k nearest neighbors are checked. Voting is done to see which of the classes are most of the k neighbors present in. The class winning the vote is assigned to the new data.
A drawback of this approach is that the value k has to be decided before hand. Checking for multiple k values can consume a lot of time and resources. In this project, the values for k were varied from 3-9 to check for the best performing results.

4.2.4 Decision Tree

As the name suggests, a decision tree model builds a tree. Each node in this tree is responsible to make a decision and every decision along a path decides the class of a data point. Each node in the decision tree checks for values of the data point on a certain attribute. Based on the value of that particular attribute, the child node to visit is chosen. The attribute present at each node is decided by how homogenetically the attribute is able to distribute classes among its children. An attribute that has a child that can predict a class with close to 100% accuracy would be preferred over an attribute whose child predicts all classes with equal probability.

Once the decision tree is built, for all test data points, decision making becomes easy. Every data point in the test starts at the root of the tree and based on the attribute suggested at that node and the value of the attribute in the data point itself, a path is taken to reach a leaf node. This eventually assigns the data point a label. Decision trees usually work best with categorical data and when the data has a large number of attributes. With the given dataset though, the number of attributes are quite low and hence it is not expected that decision trees yield a good result. Decision trees are also prone to overfitting. Overfitting is when a machine learning model learns on attributes that do not really determine the result. When new data is given to the model, it then decides accounting for these attributes that should not play any role in the decision making process. Since decision trees consider all attributes and do not disregard any, they tend to overfit.

4.2.5 Random Forest

To overcome the drawback of decision tree overfitting when learning, a new model called random forest was introduced. Random forests function in a similar way to decision trees except on two things:

- More than one tree is built.
- There is a restriction on the number of attributes that a tree can consider. This way every tree excludes ccertain attributes.

Now, when new data is passed through the model, class labels are obtained from each of these individual decision trees and the majority vote is assigned to the data point. In general it is known that of all the primitive machine learning models, random forests perform the best. In this project as well, it is expected that random forest will yield good results.

4.2.6 Multi Layer Perceptron

The latest buzzword in the industry right now is neaural networks. A neaural network has multiple layers. The very first layer is the input layer where the data is fed in. There are as many nodes in this layer as are features in the data. The very last layer in the neural network is the output layer. The number of nodes in this layer is same as the number of classes in the data. In between the input and the output layer are multiple hidden layers. Each one of these layers has an arbitrary number of nodes. The number of nodes, although arbitrary, are same in all of these layers. Every node in a layer is connected to every node in the next layer. These links between layers have weight associated with them. Every node in the hidden layer also has an activation function which manipulates the input it receives. The aim of training a neural network is to learn these weights to give out the best output. In this project, a neural network with 100 hidden layers is used. Three different activation functions are used as well:

- Sigmoid function
- ReLU
- tanh

It seems to be the case that machine learning has become associated with neural networks. However, neural networks are not always the best model to solve certain problems. Neural networks typically require huge amounts of data to learn. Since the data available in this project is quite low, neural networks are not expected to provide good results.

5 Results

All the above mentioned models were run on the data. They all were run with all possible pairs of unigrams, bigrams and trigrams. Table 3 shows the accuracy that these models achieved on various combinations of bag of words. The best result at 62.41% was obtained by the random forest model with a combination of unigrams and bigrams. The model on the other end of the spectrum was the k-nearest neighbor classifier with k set to 4 and with

a combination of bigrams and trigrams. This low was capped at 42.41%.

Table 3: Result with the accuracy of various models. The value and model in blue show the best result achieved and that in red shows the worst achieved.

	Accuracy(%)					
	Unigram			Bigram		Trigram
Model	Unigram	Bigram	Trigram	Bigram	Trigram	Trigram
SVC-L	61.03	60.69	60.34	53.45	52.76	50.69
SVC-R	61.72	61.03	61.03	54.48	53.79	50.00
RF	61.03	62.41	60.0	55.86	54.14	50.0
DT	51.03	51.79	49.34	48.93	49.34	50.24
KNN-3	54.48	53.79	53.45	51.03	51.03	50.34
KNN-4	50.69	55.52	53.45	43.1	42.41	49.66
KNN-5	51.72	54.14	53.79	45.17	44.14	50.69
KNN-6	53.1	55.17	56.55	44.48	44.83	50.0
KNN-7	53.79	54.83	54.48	45.52	43.45	52.76
KNN-8	54.14	56.9	56.9	48.62	43.45	52.07
KNN-9	53.79	58.28	57.93	48.28	47.59	52.76
MLP-l	60.14	56.66	55.79	52.14	52.1	50.31
MLP-r	59.41	57.59	55.83	52.83	51.66	50.1
MLP-t	59.62	56.62	55.66	52.59	52.21	50.31

The results shown are an average of accuracy obtained by performing a 10 fold cross validation. The performance of the models were exactly as predicted.
Analysing the results, it can be seen that the best classification accuracy has been obtained with unigrams involved and at most maybe bigrams. This makes sense since the text being considered are song lyrics and not from novels or written material. Songs do not always conform to grammatical constructs and so it is quite difficult to deduce emotions and moods by considering groups of words. Song writers also do not have the luxury of writing long sentences to put expressions to words and this further warrants shorter patterns to be recognised and used.

6 Conclusion

Avenues for future work. 0.5-1pg This project provided a way to classify lyrics into the three classes of 'sadness','tenderness' and 'tension'. There is scope for future work as well.

- The very first task would be to improve data pre-processing. It might be useful to clean the data better. It might also be possible to convert certain phrases that are understood easily by humans into more elaborate sentences. This could help improve understanding the emotion. If possible, this could be done even before annotation so that the annotators also can annotate better.
- It would be interesting to consider all the classes for the single label classifier.
- Performing classification using the multi-label dataset seems like the logical next step.
- It could also be useful to compare with more metrics other than just accuracy. Many a times accuracy does not show the real picture of the performance of the various models.
- It could also be possible to get more data from other sources and then train a neural network to classify the songs. There is a reason that neural networks are this popular.

References

Jupyter Notebook. https://jupyter.org.

LightTag. http://lighttag.io.

Pandas. https://pandas.pydata.org/.

Scikit-learn. https://scikit-learn.org/stable/.

Manash Pratim Barman, Amit Awekar, and Sambhav Kothari. 2019. Decoding the style and bias of song lyrics. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1165–1168. ACM.

Corinna Cortes and Vladimir Vapnik. 1995. Support-vector networks. *Mach. Learn.*, 20(3):273–297.

Michael Fell and Caroline Sporleder. 2014. Lyrics-based analysis and classification of music. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 620–631.

Sara Giammusso, Mario Guerriero, Pasquale Lisena, Enrico Palumbo, and Raphaël Troncy. Predicting the emotion of playlists using track lyrics.

Gongde Guo, Hui Wang, David Bell, and Yaxin Bi. 2004. Knn model-based approach in classification.

Chitralekha Gupta, Emre Yılmaz, and Haizhou Li. 2019. Automatic lyrics alignment and transcription in polyphonic music: Does background music help?

Xiao Hu, J Stephen Downie, and Andreas F Ehmann. 2009. Lyric text mining in music mood classification. *American music*, 183(5,049):2–209.

Anna Kruspe. 2016. Retrieval of textual song lyrics from sung inputs. pages 2140–2144.

Annamaria Mesaros and Tuomas Virtanen. 2010. Recognition of phonemes and words in singing. pages 2146 – 2149.

A Appendices

It might be questioned as to why Naïve Bayes was not mentioned about or considered for the classification task. Naïve Bayes is known to perform well for complete data. In fact, it would not even work on data with holes. When it comes to text and especially this project with songs, data can never be guaranteed to be complete. Naïve Bayes also performs better on numerical data rather than textual data.