Lyrics based Music Genre Classification

Dipayan Dutta
Rutgers, the State University of New Jersey
dd850@scarletmail.rutgers.edu

ABSTRACT

This project aims to build a system that can identify the genre of a song based on its lyrics. We identify a set of features that establish the style of a particular song. We curate a set of songs with five labels - Rock, Hip-Hop, Jazz, Country and Pop. Then we design three models to classify the songs into their genres - Multi Layer Perceptron for multi-class classification, Random Forest for binary classification and Convolutional Neural Networks with word embeddings. We provide a user interface which would enable a user to input the lyrics of a particular song and our program would predict its genre based on the content of the lyrics.

Categories and Subject Descriptors

[**Multi Class Classification**]: Miscellaneous; [**Deep Learning**]: Metrics—*data analysis, performance measures*

1. INTRODUCTION

Music is a way to express emotion, and because humans have wide variety of emotions, there are various music styles. It may vary from the peaceful music of Beethoven to the more modern and fast paced rap songs of Kanye West. People usually search for songs using their genre as a keyword. Also, it is very essential in Music Information Retrieval to find the similarities between the songs of same genre. Hence, classification of the songs into its respective music genres is very practical and has wide range of applications.

Songwriters deploy unique stylistic devices to build their lyrics. Some can be measured automatically and we hypothesise that these are distinctive enough to identify song classes such as genre, song quality and publication time. There is, in fact a strong evidence that it is worthwhile to look deeper into lyrical properties when analysing and classifying music. Some lyrics are for example composed in rhyming verses, and may have different frequencies for certain parts-of-speech when compared to other text documents. Further, lyrics may use slang language or differ greatly in the length (for example Hip-Hop) and complexity of the language used, which can be measured by some statistical features such as word length, and the amount of repeating text (Country songs). Lyrics are also often easier to obtain and process than audio data, and non-musicians, in particular, often rely strongly on lyrics when interacting with a music retrieval system.

However, in the field of Natural Language Processing, the classification of genres of a song solely based on the lyrics is considered a challenging task. Because audio features of a song also provides valuable information to classify the song into its respective genre. Previously researchers have tried several approaches to this problem, but they have failed to identify a method which would perform significantly well in this case. SVM, KNN and Naive Bayes have been used previously in lyrical classification research. But, classification into more than 10 genres have not been particularly successful, because then the clear boundary between the genres is often lost. So, we try to use a dataset of five genres.

Hence, we try to approach this problem as a supervised learning problem applying several methods. We analysed the relative advantages and disadvantages of each of the methods and finally reported our observations. With the advent of deep learning, it has been observed that Neural Networks perform better than the previously used models. So we designed a Convoluted Neural Network using glove word embeddings and analysed its performance.

2. DATASET

Dataset for this problem were not abundant mostly due to copyright issues. However, after comparing datasets from several sources, we found out a data set in Kaggle which was most suited for our purpose. The dataset is basically a collection of 380000+ lyrics from songs scraped from metrolyrics.com. The structure of the data is index/song/year/artis/genre/lyrics. The data was not properly structured according to our needs, like there were some songs without any genre classified to it or there were some songs whose lyrics were absent. So we had to process our data before it could be fitted to any model for classification.

Initially we had to remove some irrelevant data from our dataset, making it more compact and easy to access. Like we removed artist and song year information thus creating just lyrics and genre mapping in our dataset. Then we extracted songs of five genres - Rock, Hip-Hop, Pop, Country, Jazz. And extracted 5000 songs from each genre, making the data set practical and easy to analyse. Then we removed some songs which had very few words in its lyrics. Lyrics also

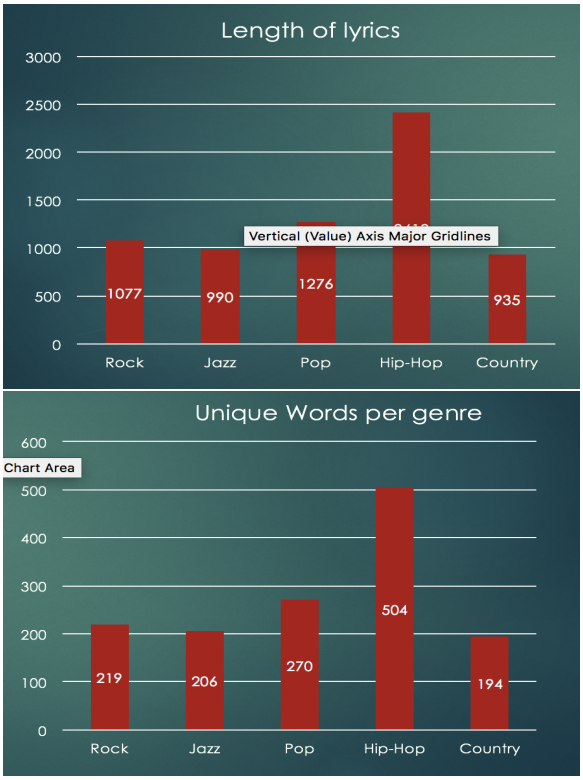

Figure 1: Analysis of the lyrics data

contained some rhyming schemes like [chorus], [verse], [x1], [x2], we removed them for simplicity. Then we tokenized the lyrics text using NLTK tool in Python. Further we applied stemming and removed punctuations. For stemming we used Porter Stemmer as we found it to be very effective.

We also did some pre-processing of data for each of our models, which would be explained later.

2.1 Data Analysis

After preprocessing we analysed the data and identified the features of data which is the first step of any machine learning problem. We used Spark to analyse the data and visualized the data. This analysis helped us understand the features of the data that would be most useful for the task in our hand.

We evaluated the average length of lyrics in each genre, and we had an interesting insight, Hip-Hop songs were longer as compared to the other genres. And the rest of the genres had almost similar lengths. Then we calculated the average number of unique words in each genre. (Figure 1) Here as well we found out that Hip-Hop songs had more unique words as compared to the rest. Then we calculated the most common words of each genre. (Figure 2) This would help us understand any correlation between the words used in lyrics and the genre type.

3. APPROACHES

We have taken three approaches to the problem, resulting in three models. In our first approach we use term frequency and inverse document frequency as our feature vectors and the genre classes as our labels to identify. We developed

Figure 2: Word Cloud for each genre

Naive Bayes, Random Forest, Support Vector machine and Multi Layer Perceptron model to classify the songs into multiple classes. In the second model we convert the problem into a binary classification problem and developed a classifier which will identify a song as Rock or Non Rock, Hip-Hop or Non Hip-Hop. We did this to identify the genres which are more distinguishable from the rest on the basis of the content of its songs. The third model that we used was the most effective of all, we used a Convolutional Neural Network, with Glove word embeddings as feature vector.

3.1 Model I

We used term frequency and inverse document frequency as our feature vectors and the genre classes as our labels to identify.

3.1.1 Bag of words

This is one of the most common approaches in text retrieval. Here, any unique term occurring in any of the document of the collection is regarded as a feature. One simple approach is to count the frequency of the word in the entire lyrical text. Another approach is term weighting scheme based on the importance of a term to describe and discriminate between documents, such as the popular tf - idf (term frequency times inverse document frequency) weighting scheme. In this model, a document is denoted by d, a term (token) by t, and the number of documents in a corpus by N. The term frequency tf(t, d) denotes the number of times term t appears in document d. The number of documents in the collection that term t occurs in is denoted as document frequency df(d). The tf-idf weight of a term in a document is computed as:

$tf \times idf(\text{t, d}) = tf(t,d) \times ln(N/df(t))$

We have also normalized the vector after applying the Count Vectorizer and Tf-Idf Weighing scheme.

3.1.2 Word2Vec

Next we used the word vectors (word2vec) to represent our lyrical text. These semantic vectors preserve most of relevant information in a text while having relatively low dimensionality. Word2Vec is an algorithm that takes every word in your vocabulary that is, the text that need to be classified is turned into a unique vector that can be added, subtracted, and manipulated in other ways just like a vector in space. We trained word vectors using python's genism library. We generated 100 dimensional word2vec embedding trained on the benchmark data itself.

3.1.3 Algorithms

With our features and labels ready we fed them into a classifier and trained it. We used 4:1 split of dataset for training and testing. We used python's scikit learn library to implement the following algorithms:

Naive Bayes: Implemented Bernoulli and Multinomial Naive Bayes

Support Vector Machine: Use the linear kernel

Logistic Regression

Decision Tree

Random Forest: Used 100 trees and majority of all the classifications is the result

MultiLayer Perceptron Model: Experimented with various activation functions and hidden layers.

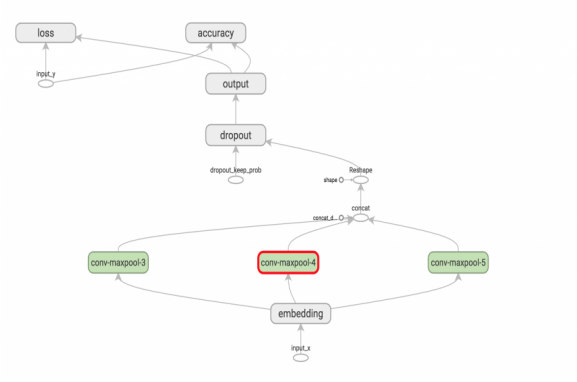

Figure 3: Convolutional Neural Network

Extra Trees Classifier: Used this algorithm to test with word2vec feature vectors.

3.2 Model II

We converted the problem into a binary classification problem and developed a classifier which will identify a song as Rock or Non Rock, Hip-Hop or Non Hip-Hop. We did this to identify the genres which are more distinguishable from the rest on the basis of the content of its songs.

3.2.1 Features

We divided the data into two groups for each of the genre classes, like grouping dataset into rock and non-rock, hip-hop and non hip-hop etc. We used one hot encoding to represent the class labels and used term frequency-inverse document frequency to represent the features. We implemented this model to identify the genres which were easily classified as compared to the rest.

3.2.2 Algorithms

We used the same algorithms as we used in the previous model for this binary classification task.

3.3 Model III

We used a Convolutional Neural Network to classify the songs into their respective genres. We used pre-trained glove vectors for this model.

3.3.1 Description of the model

The glove model we used is Google Glove 6B vector 100d. We have implemented two CNN models using Keras library:
• Simple convolution model: We have implemented a single layer of convoluted and maxpool layer.
• Dense convolution model: We have implemented multiple convoluted and maxpool layers with filter sizes of 3, 4 and 5.(Figure 3)

3.4 User Interface

After training our models, we designed an user interface, when a user can enter the lyrics of a song and our program would predict the genre of the song.

4. RESULTS

Now we report the results of experiments on these models on a dataset of 25000 songs equally distributed among all the genres.

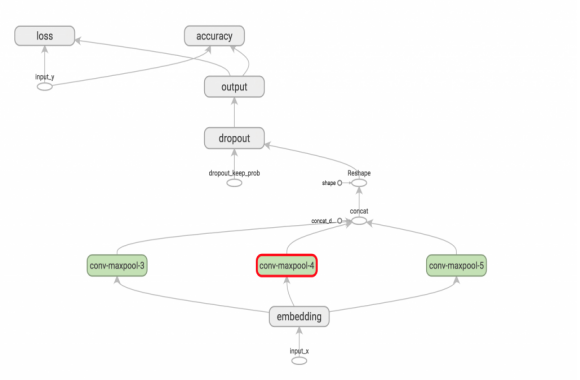

Figure 4: Convolutional Neural Network

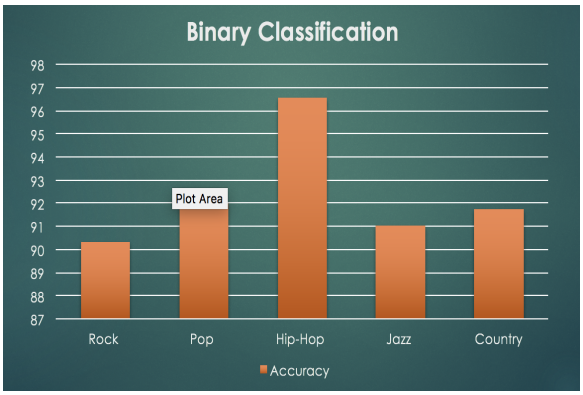

Figure 5: Binary Text Classification

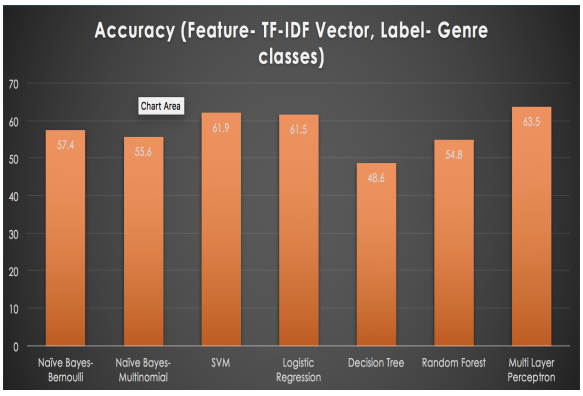

Figure 6: Multiclass Text Classification TF-IDF

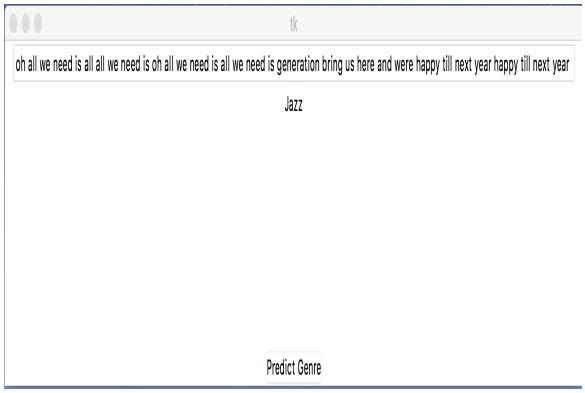

Figure 7: User Interface

4.1 Model I

A summarization of the results is demonstrated in the figure. We tested for with both TF-IDF vectors and counts as our feature vectors. We observe that TF-IDF vectors are better representation of the words in the lyrics. And among the algorithms, Multi Layer Perceptron performed better than the other algorithms with an overall accuracy of 63.5% accuracy. SVM comes close second with 61.9% accuracy. The confusion matrix shows that Hip-Hop is most accurately classified and Jazz is mislabeled most of the times. Then, we used word2vec as our feature vector, and applied the Extra Trees Classifier and Support Vector Machines, and we observed accuracy of 60.3% and 62.4%. Hence the use of word2vec did not produce significant improvement in our problem.

4.2 Model II

The results of binary classification were better, which helped us in analysing the problem in even more detail. We identified that using the words in the lyrics, Hip Hop genre was most accurately labeled as compared to the rest of the genres

4.3 Model III

We ran the model multiple number of times, changing the following parameters:

- learning rate: Modified the learning rate from 1 to 10^{-7}
- adjusting the dropout: Adjusted the dropout layers and modified its values
- modifying the filter sizes: Used filters of sizes 3,4 and 5
- Increasing the number of epochs Providing enough time for the model to learn
- Increasing batch size: Tried batch sizes of 32,64 and 128

In the simple convolutional neural network we could achieve an accuracy of 69.2% and in the dense model we could achieve an accuracy of 71%. Both were run for over a hundred epochs. This is a significant development as compared to the previous two models.

5. CONCLUSIONS AND FUTURE WORK

From the models that we developed and the experiments that we conducted we can say that the Convolutional Neural Network Model performed significantly well compared to the other models. However, the training time for an CNN is very

high even though pre-trained word embeddings were used as feature vectors. In that respect Multi Layer Perceptron, SVM and Random Forest perform well. Apart from Hip-Hop(as seen from the confusion matrix) other genres might be mislabeled at times. Accordingly, the user interface works quite well for hip-hop genre lyrics.

However limited by time, we could produce some significant results in the field of music genre classification based on lyrics. There is a lot that can be done like better preprocessing of data. Adding more data for each of genre classes. We might train a model with lyrics as well as audio features and it is expected that we can get better results. Also, we might train a more complex model which would remember order of words like an LSTM, and we can experiment on our training data.

Classification by lyrics will always be inherently flawed by vague genre boundaries with many genres borrowing lyrics and styles from one another. For example one merely need consider cover songs which utilise the same lyrics but produce songs in vastly different genres, songs which have no lyrical content. To produce a state of the art classifier is is evident that this classifier must take into account more than just the lyrical content of the song. Audio data typically performs the strongest and further research could look into employing these models to the audio and symbolic data and combining with the lyrics to build a stronger classifier.

6. ACKNOWLEDGMENTS

I would like to thank Professor Gerard de Melo for suggesting me to work on this topic. And I would like to also like to thank Mr. Rajarshi Bhowmick to have helped me in this project whenever I required help

7. REFERENCES

1. Text Classification With Word2Vec by nadbor
2. Xiao Hu, J. Stephen Downie, and Andreas F. Ehmann Lyric Text Mining in Music Mood Classification. American music, 2009.
3. Dan Yang and Won-Sook Lee Music Emotion Identification from Lyrics. Multimedia, 2009. ISM âĂŹ09. 11th IEEE International Symposium on, San Diego, CA, 2009
4. Alexandros Tsaptsinos, Lyrics based Music Genre Classification using a Heirarchial Attention Network