Text Mining and Natural Language Processing

2022-2023

Chibuzor John Amadi amadichibuuzor@gmail.com,
Kristian Perriu kperriu@gmail.com

WG?

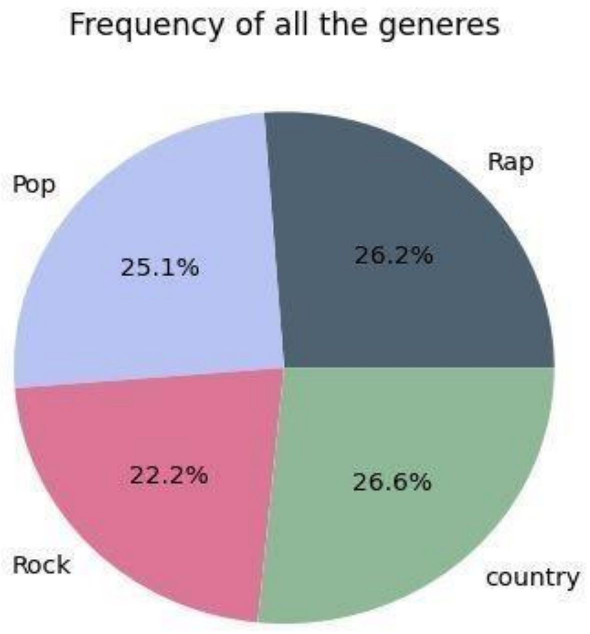
Introduction

In this project, our aim to make a classification of songs into various genres by exploiting their similarities between their lyrics. It is a recent issue we believe is present today in the music and video streaming service industry. There are cases where controversy arises in the music industry to where a song fits or what genre specific artist belong to. This arises when in cases where artist can cross boundaries and create works of art that seems to be overlapping.

A music genre is a conventional category that identifies some pieces of music as belonging to a shared tradition or set of conventions. Generally music genre are categorised by similarities between songs and sometimes by the category of the artist. The modernization of music and the modern day artist make it difficult to clearly distinguish between this genre. Another major problem could be seen in the collaborations of artist from two different genres. The Giants of the music and video streaming services like Apple Music and Spotify have developed technologies to automate this categorization process considering that they each estimate 20,000 songs each added to their site daily. (1). It is believed that they have chosen to ignore the lyrics of these songs, considering the large amounts of lyrics that has to be collected daily. However, utilizing lyrics could be useful in genre classification. (2). Spotify, Apple Music and other music streaming services use metadata such as acoustics and emotional tone for this genre classification. Aside from the fact that Lyrics of genre still differ in some aspects, but the introduction of new genres of music, such as Afro-beat, Griming, Techno and so , have a major disparity in the lyrics used but still sound very similar to the tradition music genre such as pop, rock and country.

Data

Our dataset was generated with a python module called ‘lyricsgenius’. To generate lyrics directly from the internet we subscribed to a website called Genius API, on this website we generated a token and with this token we were able to access the lyrics of artist in 4 genres; Rap, Rock, Country and Pop. After extracting this lyrics with the python module we created merge every lyric of each artist into a column and the genre of that artist and created a csv file. For each genre, we have 30 artists and for each artist we set a maximum of 100 songs to be extracted. During this extraction phase we skipped any song that was a “remix” or a “live” performance. As such the total dataset we were able to extract was 10995 song lyrics for 4 genres. We divided our data into a train test split of 80 and 20 percent respectively. the graph below shows the distribution of the dataset between the 4 gernes.



Methodology

We proposed using 4 major types of neural networks, the LSTM(Long-Short Term Memory model), the Bidirectional LSTM, the GRU(Gated Recurrent Unit) and the BERT(Bidirectional Encoder Representations from Transformers) model. The Preprocessing and architecture of the LSTMs model were very similar to configure. The preprocesssing stage began by removing a tag we noticed to appear before words (/n), we felt this as like a seperator used by the genuis api when generate this lyrics. Then we used the nltk word corpus to remove stopwords. We used Tensorflow Keras Preprocessing to carry out tokenization and lemmatization which places words in a text as tokens and the latter is put words in their original tense. After successfully carrying out this processes we then introduce the body of text to an embedding. While working with the LSTMs, we used GloVe embeddings to carryout word embeddings of the text. GloVe stands for Global Vectors for word representation. It is an unsupervised learning algorithm developed by researchers at Stanford University aiming to generate word embeddings by aggregating global word co-occurrence matrices from a given corpus.

BERT and other Transformer encoder architectures have been wildly successful on a variety of tasks in NLP. They compute vector-space representations of natural language that are suitable for use in deep learning models. The BERT family of models uses the Transformer encoder architecture to process each token of input text in the full context of all tokens before and after, hence the name: Bidirectional Encoder Representations from Transformers. How does it work? The BERT model we used is a pretrained model that we exported from tensorflow via links and tensorflow_hub.

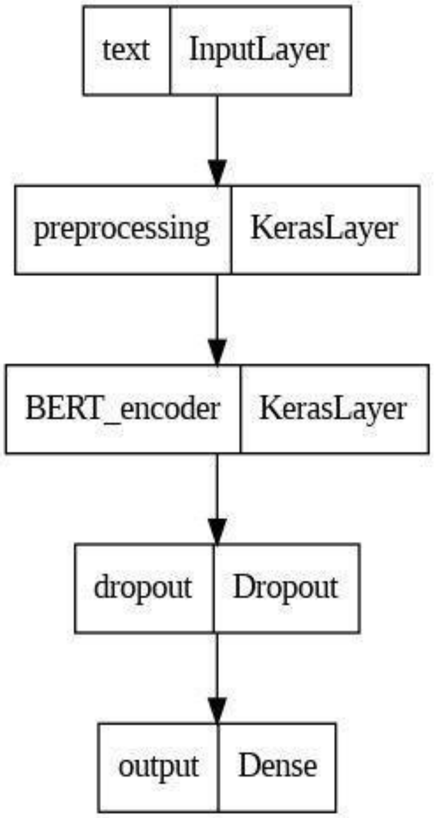
The model is pretrained in a large corpus of unlabeled data such as Wikipedia and it can then be fine-tuned for specific tasks. We then utilized the BERT transformer architecture which consists of multiple layers of self-attention mechanisms which allows the model to capture contextual dependencies between words by assigning different words different weights based on their relevance to each other which is really important in classifying lyrics of songs.

Implementation Initialization:

1. PREPROCESS-URL: This link will be used to import the preprocessor of the text. This is done because BERT by itself cannot process pure text so this link will create a model which transforms the text into multiple numeric tokens and it will also arrange them into tensors to create the initial input for BERT.

2. ENCODER-URL: This link will provide the model which will be used for the encoding and to some extent even the classification

The image below gives a graphical view of the work flow of the BERT model



Result

We experimented with 4 models for this task, the LSTM, the Bidirectional LSTM, the GRU and the BERT model. Below is a table that shows the result of each model with the accuracy metrics. For the LSTM model(s) (i.e. the LSTM, The Bidirectional LSTM, The GRU) we used the similar parameters to train them, we used 2 layers with 32 and 16 hidden layers each, we used a learning rate of 0.001, we used an activation function in the hidden layer as sigmoid and a softmax activation function for the output layer.

The accuracy we got from the LSTM Models were not encouraging and so we disregarded them and continue further with the BERT model. the accuracies can be seen in the table, Note that the accuracy of each LSTM model didnt substantially even after tuning the hyper-parmaeters. At this point BERT was the best choice.

with BERT: As described above, the raw text will go through the preprocessing model which will do the tokenization and proper arrangement and this tokenized text will be passed on the encoding model. The encoding model will do at first the embeddings for us and then it will give us a dictionary with 3 results.

What we are interested here is the pooled-output which represents each input sequence as a whole. We can think of this as an embedding for the whole lyrics the song. The pooled output is what we are going to use to make predictions later on. Each of these "models" that will do all the heavy lifting for us will be layers on our model. The architecture we decided to use was a functional one because we can pass any layer to any other layer and this allows for a very versatile architecture. Initially, the pre-processing layer will be passed to the encoding layer which will be then passed to a dropout layer to prevent overfitting. This dropout layer will then be passed to the dense layer which will carry out the genre classification

Comparison between BERT and other models.

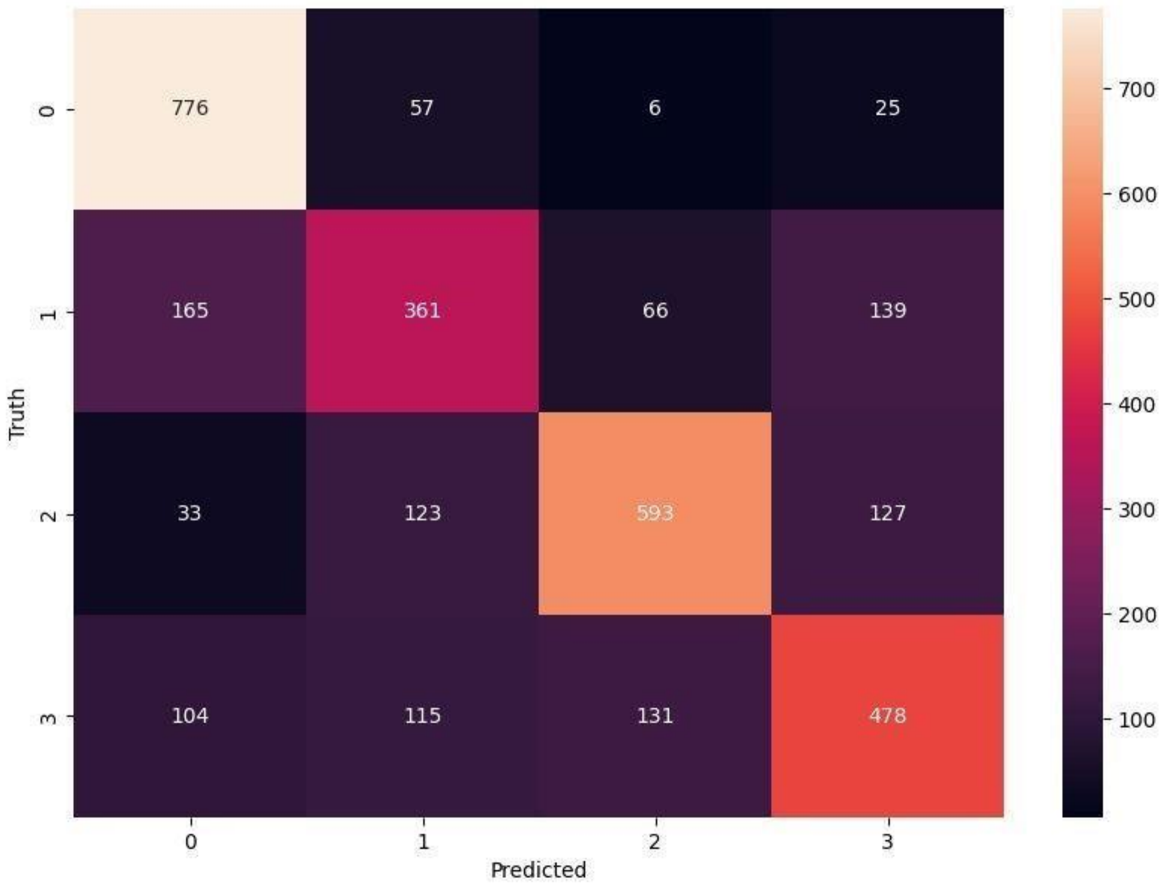
Models	Accuracy
LSTM with GloVe Embeddings	0.332
BIDERCTIONAL LSTM with GloVe Embeddings	0.303
GRU with GloVe Embeddings	0.4005
BERT	0.6101

Classification report for BERT

Label	Encoding	precision	recall	f1-score	support
Rap	0	0.90	0.70	0.79	884
Pop	1	0.53	0.45	0.49	736
Country	2	0.83	0.41	0.55	857
rock	3	0.43	0.83	0.57	822
Accuracy		-	-	0.60	3299
Macro avg		0.67	0.60	0.60	3299
Weighted avg		0.68	0.60	0.60	3299

Conclusion

When inspecting our confusion matrix, we noticed multiple instances of pop and country songs that were mis-classified as rock. We also noticed some misclassifications our model had made in pop category between rap and country but it fair to assume that these two categories bear lyrics with large similarity. Our model predicted almost accurately a clear distinction between rap and country which we honestly expected and hoped for considering a large dissimilarity between the lyrics used. Based on the classification even though it is not a 100 percent accurate we believe that with this model, if we introduce more genres of much that have more less collaborations between artist and lyrics, the result will be a lot better. Each step of the project was carried out on an equal level of participation.


776
57
6
25
165
361
66
139
33
123
593
127
104
115
131
478
Truth
0
1
2
3
Predicted
0
1
2
3
700
600
500
400
300
200
100