Comparing Logistic Regression and LSTM for Genre Classification on Song Lyrics

By: Josh Viktorov and William Ray

Abstract

Our project aimed to test and compare different models for classifying song lyrics into their corresponding genre. We compared a simple logistic regression model, with a more sophisticated LSTM to see if context would improve accuracy over a bag of words model. After comparing, we were able to conclude that neither model was significantly better than the other, and therefore context may not be crucial for genre classification.

Introduction

The genre of a song is often determined by the style and content of the song. Beyond that, songs of the same or similar genres invoke similar emotions in listeners. With a tremendous amount of music on the internet, understanding which genres apply to which music is an important task for recommendation systems and information retrieval. Searching by genre can be a helpful tool to find more relevant songs faster. Understanding how different genres are defined can also provide useful cultural insights. Additionally, understanding a genre's essence can help artists compose music to satisfy a certain demographic best. While there are many features to observe when classifying songs into genres, such as tempo, instruments, and amount of vocals, lyrics lend themselves to this task. Lyrics are composed entirely of text and there exist many techniques to classify text. Additionally, lyrics are well suited to most information retrieval tasks. It is far easier to search for the words of a song than to describe what the song sounds like or what it feels like.

There are many ways to represent text in a way that can be effectively processed. A document can be represented as a bag of words, a sequence of n-gram encodings, or even a sequence of word-level encodings. Furthermore, there are many options when choosing how to process text represented in this way such as Naive Bayes, logistic regression, or advanced recurrent neural networks models such as LSTMs. We wanted to explore how the inclusion of context affected a model's ability to classify song lyrics by genre. Understanding the importance of context in this situation may generalize to other text classification tasks. To understand this, we trained a logistic regression model on a bag of words representation of the song lyrics as well as an LSTM model on a sequence of word-level embeddings. The embeddings were learned by the LSTM that was trained on the dataset we used, and we did not use pre-trained word-embeddings. These models were chosen because the LSTM model has a notion of what order the words appear in the song lyrics whereas the logistic regression model is working with a representation of the song that loses all sequential information. We found that the LSTM model was able to perform slightly better than the logistics regression model, but the difference was not very significant.

Related work

Another approach, Music Genre Classification by Lyrics using a Hierarchical Attention Network[1], applied a Hierarchical Attention Network (HAN) to the task of classifying song lyrics by genre. The HAN was composed of a bi-directional gated recurrent unit that functions similarly to the LSTM that we used. The hierarchical network attempted to replicate the structure of the lyrics and was able to learn which sections, lines or words impacted genre classification the most. This work also classified songs into 20 different genres, far more than other works. The paper found that their hierarchical attention network was able to outperform basic non-neural methods.

The paper Lyrics-based Analysis and Classification of Music[2] experimented with novel approaches to three classification tasks using song lyrics. The paper explored genre detection, distinguishing the best and worst songs, and determining the approximate publication time of a song. The researchers were able to design different feature classes that reflected the song's stylistic and linguistic dimensions. The features they were able to extract were vocabulary, style, semantics, orientation towards the world, and song structure. Through experimentation, the researchers found that lyrics may play a role in all three classification tasks. Interestingly this paper found that on the task of genre classification, rap was the easiest genre to detect. In our dataset, the hip-hop genre contained songs that could also have been classified as rap and hip-hop was the easiest genre to classify compared to rock and pop. This suggests rap/hip-hop are significantly different from other types of music.

Finally, the paper Overview of Automatic Genre Identification[3] provides insight around document genre classification. The paper discusses three methods for automatic genre classification. The “traditional” method it describes as extracting features from a document and then applying any type of classification algorithm. The paper notes that often extracting good features is the most important part and that several classification algorithms can work well if given suitable features. The “character-based” method for genre identification simply represents the document as a sequence of characters. The paper describes this method as easy to implement as there is no need for manual feature extraction. Text-based documents are essentially already represented as a sequence of characters. Lastly, a less common “visual” method to genre classification is discussed, where a document is represented as a bitmap. The application of this method is typically in the case of scanned documents and information retrieval systems in an office environment. The paper concludes by saying that most genre-identification tasks can be accomplished with "traditional” methods.

Data

We decided on using a Kaggle dataset that included 227,564 different song lyrics over six different genres. Three of these genres, however, were more niche, Brazilian-based genres and we wanted to focus on our project on just American songs so we took out the songs that were those genres. This left us with Hip Hop, Rock, and Pop as our three genres, and 168,693 songs. We also noticed there were many songs in different languages, and so we parsed out all of the non-English songs as well. This left us with 123,987 different total songs. Our dataset was split into two .csv files. The first included all of the lyrics, with their associated artist, link, song

name, and idiom (language). The second .csv file included every artist and their associated genre. Combining these two files and stripping out all of the unnecessary genres left us with a file containing just lyrics and their associated genre.

Method

We used two different models to handle the problem. The first was a bag of words logistic regression model. To implement the logistic regression model we used scikit-learn. To create our bag of words representation, we used scikit-learn's CountVectorizer() function and also passed in the parameter of stop_words=english. This made the model ignore any words like "the", "is", "are,", etc. so that the model would only focus on content words. We also encoded the labels as one-hot vectors. So that "Hip Hop" was turned to [1, 0, 0], "Pop" to [0, 1, 0], and "Rock" to [0, 0, 1]. For our test/train split we used scikit-learn's built-in test_train_split() function. This broke up our data into a test sample of .4, which translated to 49,595 samples, and a train sample of .6, which translated to 74,392 samples. We then ran our model on the data to get the weights vector as well as the accuracy, precision, and recall for each genre.

The other model we trained to classify song lyrics by genre was an LSTM model. The model was implemented in Keras and consisted of a 300-dimensional embedding layer followed by 3 LSTM layers, each followed by a Dropout layer with p=0.25. Next was another dropout layer with p=0.5 before a three-neuron Dense layer with softmax activation. The model was compiled with a categorical cross-entropy loss and RMSProp optimizer. The labels were encoded the same for both the logistic regression model and the LSTM model.

We found this model to be more accurate than a model with similar architecture but instead of an embedding layer, the model was fed character-level one-hot vectors as time steps. We assumed the sequential nature of song lyrics to be important to their classification and thought a model that can identify patterns in a series of data would perform well on this task.

Results

After running the logistic regression on the training, we were left with the weights for each word in the corpus. This allowed us to pull out the top 10 highest weighted words that the model determined for each genre. The results are shown below.

Hip Hop	Pop	Rock
soulja	akon	bury
tryna	tryna	disease
ashanti	chipmunks	kneel
nigg*z	konvict	99
usher	nicki	shove

dogg	transcribed	brian
nigg*s	b*tches	roam
plus	i'ma	zevon
i'ma	conceal	iron
relationship	gaga	undead

In the progress report, we noticed that the logistic regression was picking out artists' names in the lyrics and then giving those the highest weights as they would obviously give insight into the genre. After looking more closely at our data, we figured out that what was happening is that it was taking these names from feature listings and markers of who is talking. For instance [Verse: Eminem] before his verse. To fix this issue we applied regex to the dataset to strip out any text within brackets, as this is how the data is formatted throughout. This didn't work perfectly but a larger portion of the unwanted text was parsed out. One thing to notice is even after parsing out most of the mentions, there are still some artist names in the highest weighted words ("akon", "nicki", etc.). After examining the data, we found that a large portion of these names were actually within the lyrics themselves, and the artist was just saying either their own name or the name of another artist.

The results we got confirmed our hypothesis going into the project that the logistic regression model would give us some interesting insights into the genres and what they culturally reflect. For instance, we can see how swearing is prominent in some genres like Hip Hop and Pop however may not be as prominent in Rock. Rock's top words also all tend to have similar connotations. Words like "undead", "iron", "disease", and "bury" all seem to focus around a dystopian, dark, almost negative theme. It is not the purpose of our paper to try and do any cultural analysis however it is an interesting insight, and perhaps one that could be expanded on in the future by others.

One final thing to take from this data is that in the pop section, the word "chipmunks" is the third-highest word. This is because our dataset had the entire discography of the Alvin and the Chipmunks. The fact that this had such an impact on our results suggests that one drawback to our data is that it isn't large or diverse enough and in the future may need to be expanded.

After running the trained model on the test set, we were able to gauge how well this model actually performed. The first thing we looked at was the overall accuracy of the data. We were able to get an accuracy of 0.67. While not incredibly high, this overall accuracy is still respectable. This accuracy didn't change from when we ran it for our progress report, which was to be expected since the only improvements we made were to parse out the words in brackets.

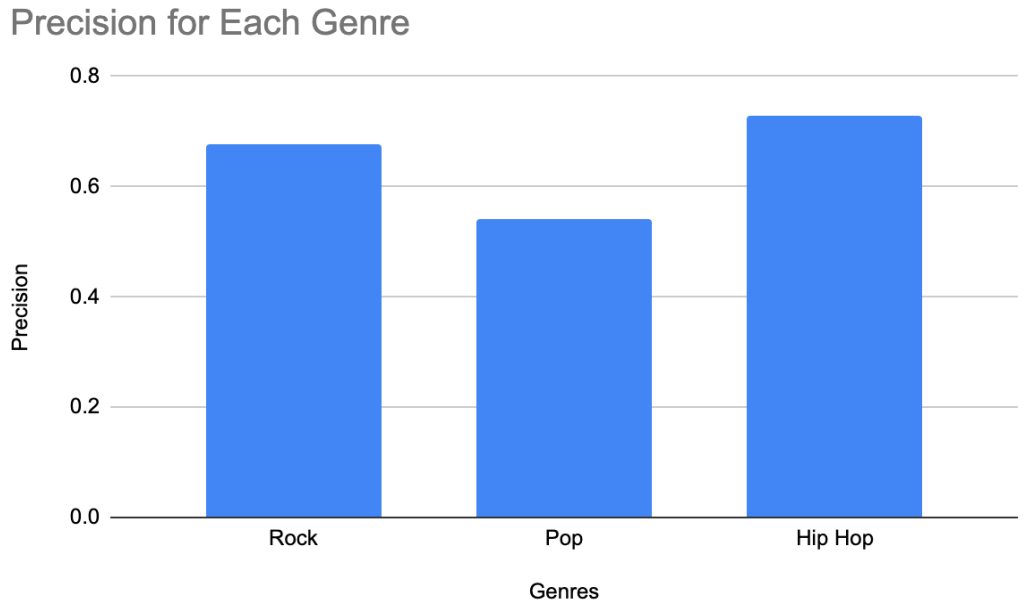
In addition to the accuracy, we got the confusion matrix of the logistic regression using scikit-learn's confusin_matrix() function. The result is shown below.

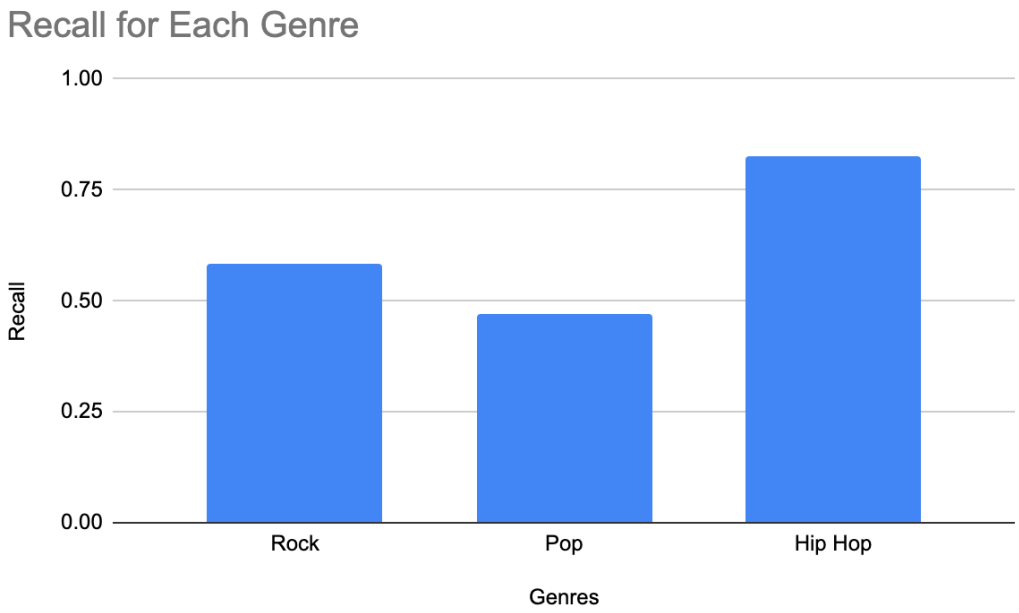
	Rock	Pop	Hip Hop

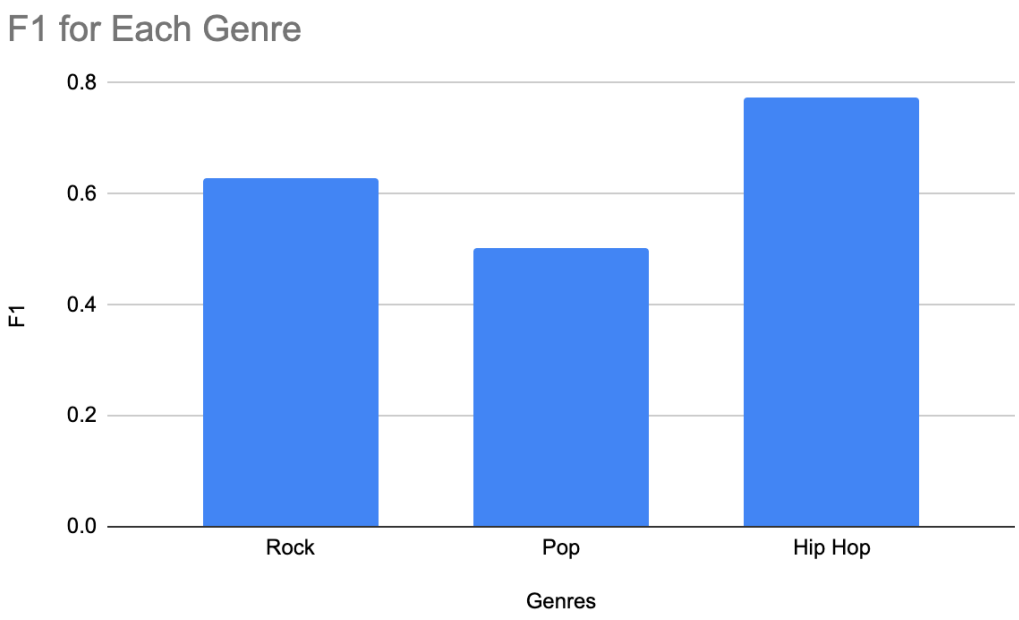
Rock	2667	1354	539
Pop	1052	3799	3223
Hip Hop	230	1873	10058

The confusion matrix gives a more detailed view of how the model performed. Each row represents the true labels, while the columns represent what the model predicted. Immediately it is clear that Hip Hop is where the model performed best. With the majority of its guesses at the intersection of the true and predicted label, we can see that the model accurately predicted Hip Hop at a high rate and miscategorized it at a low rate. We also can see that of the three genres, Pop and Hip Hop are the most similar. This is seen by looking at the intersection of the Pop column and Hip Hop row, and vice versa. These two spots have a very high count indicating that the model often mixed up the two. The fact that Hip Hop and Pop are similar can also be seen in the logistic regression top-weighted words. Both Hip Hop and Pop share the word “i’ma” as well as “tryna” as a word with a high correlation to the genre. Similarly, we can see that Hip Hop and Rock are very dissimilar. There were only a couple of hundred lyrics in which the model mixed up the two.

For a more quantitative analysis, we used scikit-learn’s precision_score(), recall_score(), and f1_score() functions to calculate their respective statistics for each genre. Below are the graphs of the three genres compared scores in each statistic.



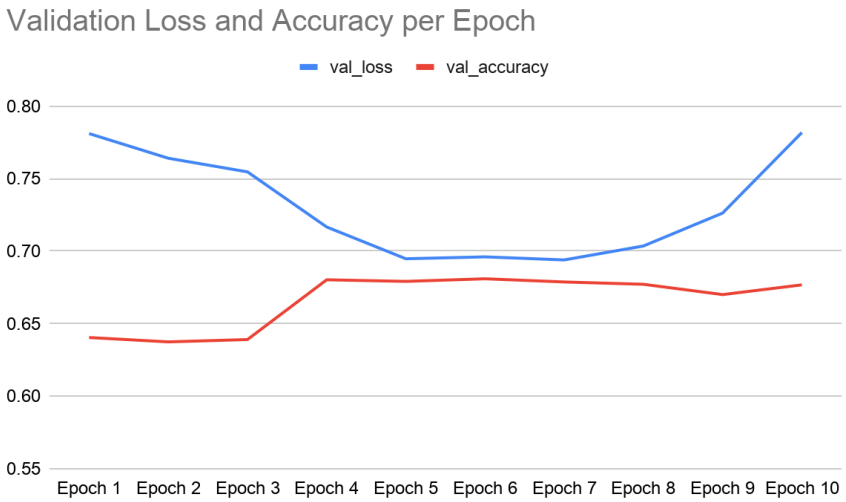




One result, which corroborates what was observed from the confusion matrix, is that the model performed best with Hip Hop in every category. Its recall and F1 are a great deal better than the other genres, however, its precision is only slightly better. This may suggest that the model was actually over guessing Hip Hop, while under guessing Rock and Pop. There could

be several reasons for this that would need to be explored in further research. Another thing to note is that Rock outperformed Pop in every category as well.

Along with our logistic regression model, we also ran an LSTM model. We trained the model for 10 epochs. The graph below shows the validation loss (in blue) and the validation accuracy (in red) over each epoch.



Over the first couple of epochs, the loss dramatically decreases as well as the accuracy increases. This begins to stall however around epoch 4, where the accuracy and loss level off. Then around epoch 7, we can see the model begin to overfit, and the loss begins to go back up to around .8. Our best accuracy was reached at epoch 4 at .68. The loss at epoch 4 however wasn't the best loss, and instead, the best loss came at epoch 7 at .694.

Discussion and Future Work

For our project, we set out to see how different models would perform on classifying lyrics' genres, as well as see what kind of conclusions we could draw from our results. After testing the different models, one interesting result we found is that the LSTM didn't perform that much better than the logistic regression model. This may suggest that certain words just by themselves are enough to classify lyrics by genre. This confirmed our thoughts going in. This may be because different genres of music tend to stick to the same topics. Looking at what the logistic regression picked out as the important words further supported this conclusion, with the words being what would be expected of those genres. A word like 'guitar" or "whiskey" will be heavily correlated with Rock, regardless of the context in which it was said.

In the future, there is still much more that can be done with this project. The most straightforward next step would be to just add to the dataset. There are several different areas where the dataset can be improved. The first is in the variety of artists. This dataset was created in a way such that for every artist, it has many of their songs. This means that while having lots of songs, a model could still be over-fit to a certain number of artists and if the model is given an

artist it hasn't seen before, perform poorly. To remedy this, in the future a greater variety of artists could be added, and instead of adding lots of songs for each artist, just adding 10-20 to not overcrowd the data. Another way to improve the dataset in the future would be to add more genres. Right now we just have it trained on 3 genres, however, there are many more genres that can be done, as well as even more subgenres. This could be done by either finding another pre-existing dataset or creating our own. The final improvement on the dataset in the future would be creating a more sophisticated way of labeling lyrics to a genre. Right now the way it works is every song has an artist, and that artist has an associated genre. This works fairly well however many artists often go between genres and so it might make more sense to do it on a song-by-song basis. This would require lots of human annotations however and is of low priority in the future.

Finally, as well as improving the dataset, another thing to do in the future is to improve and add to our neural network model. This includes both testing different hyperparameters with our current LSTM, as well as trying other models to see if they perform any better. The next model to try and use for this task would be BERT. This would be done both using fine-tuning as well as using BERT for feature extraction by taking the CLS embedding for logistic regression.

References

[1] A. Tsaptsinos, “Music genre classification by lyrics using a hierarchical attention networks,” 2019.

[2] M. Fell and C. Sporleder, “Lyrics-based analysis and classification of music,” 2014.

[3] M. L. Joˇzef, “Overview of automatic genre identification 1,” 2007.