Song Genre Classification using Lyrics

Shruti Sinha shruti85@seas
Hemanth Vihari Kothapalli hkot@seas
Christopher Fischer cdf@seas

Abstract

This project is an attempt to tackle the problem of song genre classification based solely on the lyrics. Song lyrics, specifically, exhibit properties different from traditional text documents - often containing rhyme patterns, unconventional distributions of parts of speech, common use of slang, and frequent repetition of words. Here, in this project, we aimed at identifying those set of conventions or patterns in songs and through such features, identify the genre of a song using machine learning algorithms. We utilized a data set of 362,000 song lyrics obtained from *metrolyrics.com*. The models were trained on a sample of 14,000 songs for each of the five genres: Hip-Hop, Metal, Pop, Rock and Country. The strongest resulting model is capable of classifying the song genre correctly 65.2% of the time on unseen lyrics.

1 Introduction

Genre classification is of great relevance in the field of music. Accurate genre classification can aid in the effectiveness of music recommendation engines, help unearth similarities between different music genres, and reduce the need for the hand labeling of genres in streaming services. The popular approach used to classify song genres is to analyze music on the audio level. However, here we identify genres, based solely on lyrics. We extract various textual features from our data set and test our models on different combinations of these features, in order to improve overall performance.

2 Related Work

There is a fair amount of literature about song genre classification using different features, however, all those picked songs based on artists. Wang compared the predictions of Naive Bayes, SVM and Neural Networks to identify the artist of an input songs lyrics from a given set of rappers. Their highest performing model was the set of one-vs-all SVM, using the Norm feature extractor (A Wang, 2017). Xiao Hu in his paper on usefulness of text features for music mood classification concluded that bag of words approach for lyric features outperformed audio features in categories where samples are more sparse or when semantic meanings taken from lyrics tie well to the mood category (Xiao Hu, 2009). R. Mayer created a feature set combining rhyme, part of speech and simple text statistic features and proposed that this feature set performed at par with the standard bag of words approach, at baseline with the combination of both - outperforming bag of words only approach (R. Mayer and Rauber, 2008). They achieved 33.4% accuracy when predicting 10 genres.

3 Data Set and Features

3.1 Data Collection and Preprocessing

The data set that was used throughout this study was provided by metrolyrics.com and contained 362,000 song lyrics of different genres. The distribution of songs per genre can be seen in Figure 1. During the first steps of basic pre-processing, we removed incomplete songs (eg. missing lyrics or data about genre, artist or year information). We removed non-English songs, as all tokens would be lemmatized in later steps. Also, we removed songs for artists with fewer than 5 songs, as we wanted data to be representative of their respective artists and genres. As a result we had a remainder

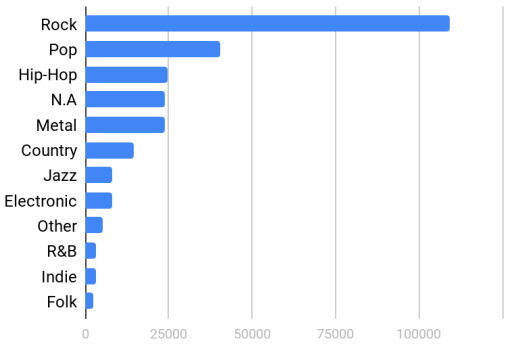

Figure 1: Data distribution

of 266,556 song lyrics (note that 92% of the data dropped was a result of missing fields). On this reduced data set, we then performed further preprocessing, using the following steps:

1. Character case to lower for all song lyrics
2. Used Twitter tokenizer from the NLTK library to preserve conjunctions and also as we wanted to observe the affect of conjunctions on our results.
3. Cleaned tokens, expanded conjunctions (eg. "I'm" to "I am")
4. Removed stop words and digits
5. Lemmatized token to present tense (eg. "cries", "cried", "crying" all refer to the same action and are thus converted into "cry")

The example depicting our tokens is shown in Figure 2. To ensure equal distribution across genres, a sample size of 14,000 per genre were taken from the top five represented genres (Metal, Hip-Hop, Country, Rock, and Pop).

Figure 2: Cleaned tokens

3.2 Feature Extraction

The features are determined by the vocabulary of songs. For feature generation, we employed parts of speech, bag-of-words, rhyme and textual statistic features.

3.2.1 Textual Statistic Features

Text documents can be described by simple statistical measures based on word or character frequencies. Measures such as average length of words in a song, unique word density can give an indication of complexity of text and is expected to vary over different genres. Fourteen such statistical features were extracted as shown in the table (Table 1). The counts of punctuation may differ and be significant in some genres, like hip-hop. We additionally calculated curse word density, with the results matching our expectation, as hip-hop had the highest curse word density.

Number	Feature
1	Total count of words
2	Number of lines in a song
3	Average line length
4	Average word length
5	Total number of unique words
6	Unique word density
7	Average unique words per line
8	Number of contractions
9	Number of digits
10	Contraction density
11	Digit density
12	Number of punctuation
13	Total curse words
14	Curse word density

Table 1: Overview of textual features

3.2.2 Parts Of Speech Feature

Part of Speech tagging is a lexical categorization or grammatical tagging of words according to their definition and textual context they appear in. Different categories include nouns, verbs, adjectives or articles. We presume that different genres will differ in the category of words they are using as well. We get a total of 45 parts of speech features.

3.2.3 Bag-Of-Words

This is a common approach in text retrieval. We construct a lexicon, the union of the set of tokens present in each song, and each unique term

becomes a feature. We can filter out uncommon words to prevent overfitting and calculate the relative term frequencies (TF) of each word in the lexicon for each training example. To prevent extremely common words like the and is from having a higher weight from more uncommon, but possibly more semantically significant words, we multiply Term Frequency with the inverse document frequency (IDF). The main draw back of this feature generation is that the word ordering is lost. The TF-IDF weight of a term is calculated as:

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

3.2.4 Doc2Vec Features

Further, we used a very different approach to feature generation through the learning of a fixed length vector. Doc2Vec is based off of the well-known Word2Vec model and employs a continuous bag of words approach that implicitly makes the assumption that nearby words have some semantic link. It generates fixed length dense vectors that encode semantic meaning of a document. A vector size of 250 was chosen based off previous literature and preliminary experiments.

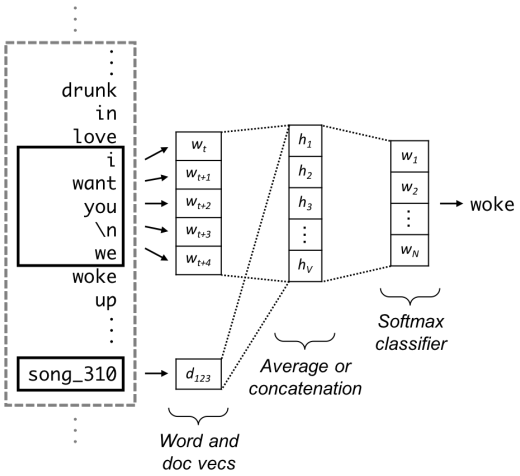

Figure 3: Doc2Vec

3.3 Models/Methodologies

We performed experiments on three different models: Naive Bayes, Logistic Regression and Decision Trees.

3.3.1 Experiment 1

We split our data set randomly into training and testing sets and performed four experiments using combinations of different features:

1. *Only Textual Statistic Features*
2. *Textual + Parts of Speech (59 features)*
3. *TF-IDF (uncleaned tokens)*
4. *TF-IDF (clean tokens)*

Note that textual features include the Doc2Vec 250-length vector. We tested these different feature sets on each of the three models.

3.3.2 Experiment 2

To test our generalizability of our models, we applied them in two transfer learning tasks across two domains: artists and years. Figure 4 shows the distribution of our data, grouped by year. Clearly there is a skew in when the songs collected were released, with years 2006-2007 making up approximately 50% of our data.

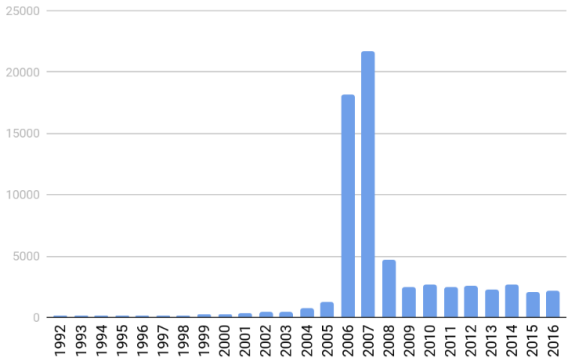

Figure 4: Metrolyrics data set year distribution

Figure 5 shows a histogram of the number of songs each artist has. An exponentially higher number of artists only have a few songs. Though notably, 97% of all artist's songs are completely classified under a single genre. To obtain train and test splits, we did a random sample across artists that ensured a 60-40 split. Seeing that artist and genre are so closely linked, such a sampling technique should still ensure a reasonable representation of each genre within each set.

For both of tasks, the best model from Experiment 1 was taken. This was the logistic regression trained on cleaned TF-IDF tokens.

3.4 Results

3.4.1 Experiment 1

Without restricting the depth of trained decision trees, we observed severe overfitting, giving an train accuracy of 99%. So, we restricted its depth to 15 for dense features and 25 for TF-IDF and observed a drop, with TF-IDF giving a train accuracy

	Textual	Textual + POS	TF-IDF	TF-IDF (clean)
Decision Tree	49.5	51.3	50.1	50.2
Logistic Regression	**52.9**	**54.5**	**61.7**	**65.2**
Multinomial Bayes	48.7	50.4	59.4	62.3

Table 2: Experiment 1 test accuracies

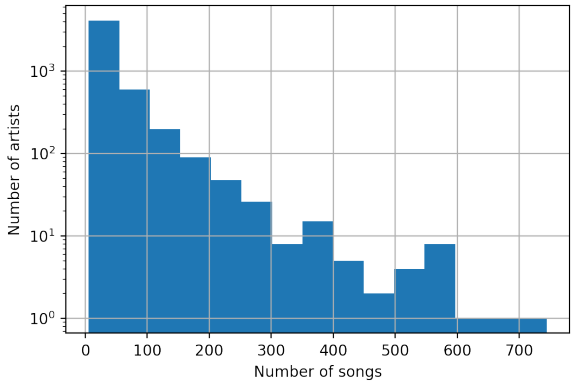

Figure 5: Metrolyrics data set artist histogram

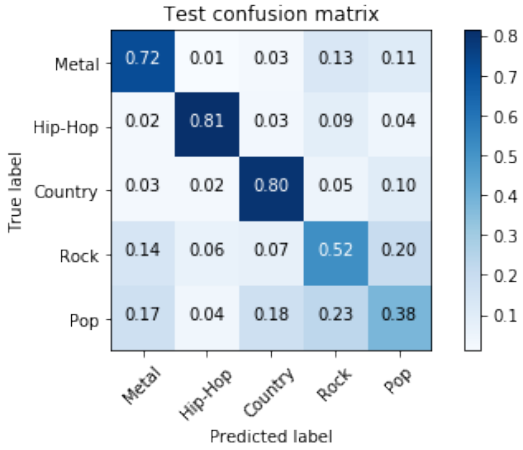

Figure 6: Experiment 1 confusion matrix

of 81.6%. On the other hand, Naive Bayes exhibited high bias and was unable to achieve a high accuracy on the train set for all sets of features. The best accuracies was reported consistently by using cleaned TF-IDF. Overall, logistic regression performed best, giving a test accuracy of 65.2% on TF-IDF (clean) feature set.

The confusion matrix in Figure 6 shows that Hip-Hop, Country, and Metal had the top three accuracies respectively.

Figure 7 shows the relative importance of each textual feature. This was measured by removing each feature one at a time and finding the resulting test accuracy (which can be seen on the vertical axis). A more important feature is then one that causes the largest drop in accuracy. By this measure, *number of lines*, *number of punctuation*, and *doc2vec* were the most important. Note that the baseline, with all features present, is 54.5% (as seen in Table 2).

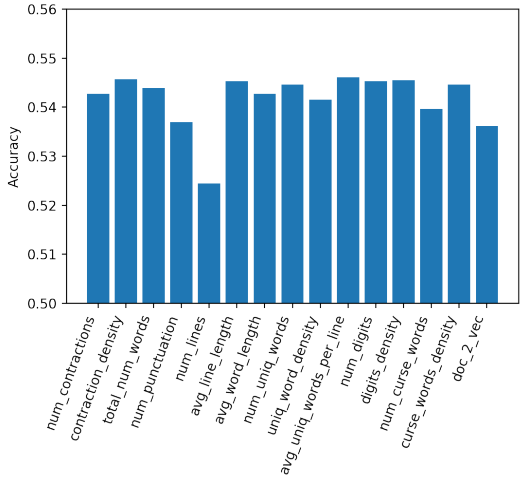

Figure 7: Experiment 1 feature importance

3.4.2 Experiment 2

In Figure 8, the train and test accuracy resulting from splitting train and test at each year from 2000 to 2015 can be seen. Overall, these accuracies are lower than those achieved in Experiment 1. The general increase in test accuracy is justified, because as the split moves closer to present day, the train set gets larger and the test set gets smaller. This is likely also the reason that the train accuracy decreases, though to a smaller degree. The spike in 2006 accuracy is likely due to the skewed data set, with a majority of the songs being pulled from 2006 and 2007. Figure 9 shows the confusion matrix for splitting on year 2007. While the top performing categories remain the same, we do see a larger drop in metal than other genres, though pop accuracy increased.

For transfer learning on artists, an accuracy of 62.3% was achieved. This is notably higher than the year accuracies, though lower than what was achieved in experiment 1.

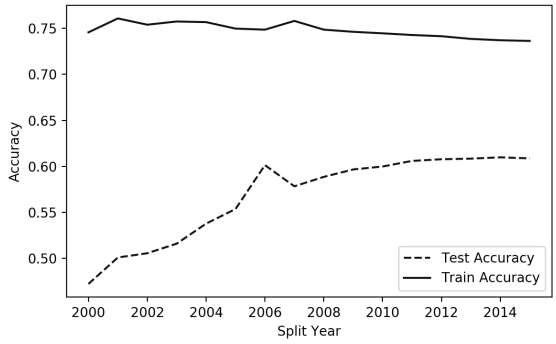

Figure 8: Experiment 2 accuracy at each yearly split

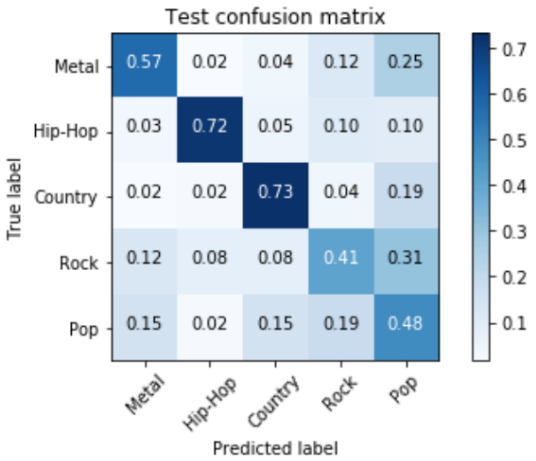

Figure 9: Experiment 2 confusion matrix, split on year 2007

3.5 Conclusions

Looking at the test matrix for experiment 1, the higher accuracies in hip-hop can be explained by the stark difference in textual features from all other genres. Hip-Hop is nearly always the outlier in its songs' length, number of punctuation, and curse word density for example. Furthermore, these textual features were also measurably contributed more to accuracy than any others, sans doc2vec. Similarly, along with metal and country, it uses a comparatively unique lexicography. Manually looking at the most frequent tokens in rock and pop, on the other hand, reveal that their vocabularies are remarkably similar. It is not surprising then that the most common mistakes make were confusing rock with pop songs and vice-versa.

From experiment 2 we can conclude multiple things. Given the drop in accuracy going from a random test train split to a split based on the year, there are likely changes to the genres vocabulary over time. Such a change in either the frequencies of words or the actual words themselves would directly lead to a decrease in the applicability of the trained TF-IDF model. When splitting by artist, although the accuracy did decrease, it was higher than when splitting by year. It is plausible that the lexicography between artists of the same genre is more similar than the lexicography between years of the same genre.

We have developed and tested multiple classification models to identify the genre of an input of lyrics based on vocabulary and textual features. Data preprocessing was tuned to filter out noise from tokens resulted in an increase in accuracy. The highest performing model was the logistic regression, using the cleaned TF-IDF feature set. This achieved a train accuracy of 73.6% and a test accuracy of 65.2%. Interestingly, decision trees performed the worst on token features, likely because it is a more complex model and easily overfits to the training data and thus fails to generalize well. Throughout all experiments, we find the key to achieving a high accuracy is finding features that are not only consistent across songs in the same genre, but different from those in others. We believe that our results are comparable, if not better, than those achieved by Mayer, even through the differences in our data sets.

3.6 Future Work

Seeing how closely linked genre and artist are, future work would primarily involve applying the techniques described here to artist classification.

References

V Ranganathan A Wang, R Cheong. 2017. whose rap is it any- ways? using machine learning to determine hip-hop artists from their vocabularies".

R. Neumayer R. Mayer and A. Rauber. 2008. rhyme and style features for musical genre categorization by song lyric". *Proceedings of the International Conference on Music In- formation Retrieval*.

Andreas F Ehmann Xiao Hu, J Stephen Downie. 2009. lyric text mining in music mood classification.