Lyrics Meet Melody: A NLP Approach to Music Genre Classification

November 10, 2023

Marti JIMENEZ
m.jimenez@student.utwente.nl

Samuel COSTE
s.f.coste@student.utwente.nl

Abstract

This paper presents a comprehensive study on the application of natural language processing techniques for music genre classification based on song lyrics. Focusing on the intricate relationship between textual data and genre categorization, the research aims to address the challenge of accurately predicting music genres using only the song lyrics as input. Leveraging a dataset comprising song lyrics with corresponding genre labels, the study explores the effectiveness of Word2Vec embedding models in representing textual data numerically. The developed feed-forward neural network employs a hidden layer configuration with tanh activation functions, emphasizing the intricate interplay between model architecture and feature representation. Through a meticulous evaluation process utilizing key performance metrics, including accuracy, precision, recall, and F1-score, the study provides insights into the model's predictive capabilities and limitations, achieving an accuracy of 61.8% for 6 different classes. The findings underscore the significance of balanced feature engineering and highlight the complexities associated with classifying music genres solely based on textual data.

1 Introduction

The contemporary music landscape is flooded with an unprecedented array of genres and styles, each offering a unique listening experience. As music streaming platforms continue to grow, the need for effective genre classification becomes increasingly vital for ensuring tailored recommendations and an enhanced user experience. In light of this, our project delves into the challenging realm of music genre classification, focusing specifically on the intricate task of predicting a song's genre based solely on its lyrical content.

Drawing inspiration from the rich possibilities of Natural Language Processing (NLP), we have set out to investigate whether it's possible to decipher a song's genre purely through the analysis of its textual components. While humans might intuitively recognize (or not) the distinctive elements that define various musical genres, teaching a computer to do the same presents a complex and multifaceted problem. Our endeavor is to bridge this gap by exploring innovative NLP techniques and machine learning models, aiming to unlock the potential for accurate and reliable genre prediction solely from the lyrical data of a song.

Through careful data collection, preprocessing, and feature engineering, we seek to uncover the underlying patterns and nuances within song lyrics that contribute to genre classification. Our project not only seeks to push the boundaries of computational understanding but also aims to shed light on the intricate relationship between language and music, unraveling the interplay between textual content and musical genres.

2 Related Work

Drawing inspiration from the "Music Genre Classification using Song Lyrics" research project conducted by Anna Boonyanit and Andrea Dahl at Stanford, our project was motivated by the challenges posed in accurately predicting music genres solely based on song lyrics. The Stanford project explored the complexities inherent in defining music genres and sought to improve genre classification accuracy by leveraging a combination of word embedding techniques and neural network models. Specifically, the use of LSTM architecture and various embedding methods, such as GloVe and Word2Vec, offered valuable insights into the role of textual data in genre classification tasks.

We recognized the potential for achieving higher accuracy through a simplified approach that could effectively capture the nuances of genre-specific language in song lyrics. By streamlining our feature engineering process and focusing on key aspects of the lyrical content, we aimed to enhance the predictive capabilities of our model while addressing the challenges posed by genre classification intricacies.

3 Data

For this research project, we used the 'Genius Song Lyrics' dataset available on Kaggle. This dataset was last updated in 2022 and was scraped from the 'Genius' website, a place where people can upload annotated works (mostly songs). This dataset is an extension to the '5 Million Song Lyrics Dataset' lyrics dataset. Every entry in this dataset has a mother tongue assigned to it. Even though the dataset mostly contains songs it also has some other types of entries such as books and poems, but these were ignored for the purpose of our research. Each entry had the following attributes:

Tag	Explanation
title	The name of the song
tag	The genre of the song
artist	Artist to whom the work is attributed.
year	Year of publication.
views	The number of page views.
features	Lists other artists that contributed.
lyrics	The lyrics of the song
language	Main language of the lyrics

Before being able to use this data we needed to clean it, which we will discuss in section 4.1.

4 Method

4.1 Data Processing

Before being able to train our model we had to process our data. As the dataset contained more than only songs entries we first had to filter out every non-song entry, which we could do based on the 'tag' column. Because the vast majority of the dataset contained English lyrics we decided to use only English for our research, as mixing other languages into our data would most likely produce mainly noise. We standardized the text by removing everything which were not lyrics in the 'lyrics' column. Looking into the lyrics manually we found different kinds of annotations that we filtered out using regex. This was done to preserve the essence of the research which is to classify solely based on the lyrics. Now that we have clean English song lyrics we standardize the text by converting it to lowercase and tokenizing it using the NLTK libraries. All duplicate words were removed from the lyrics to mitigate any skewing effect frequent words could have within a song.

The approach to turn the raw text into something useful for our neural network was to convert the complete song into one vector. In order to do so every word was vectorized using the word2vec-google-news-300 model. This model takes a word as input and returns a vector of size 300 that embeds the meaning of that word. By applying this vectorization to every unique word of the song, then taking the average of all the vectors before normalizing the vector to contain all values between zero and one, a single vector of size 300 is created that reflects the average 'meaning' of the song. It is this vector that is then used as input to train our neural network model.

4.2 Feed Forward Neural Network

Our project employs a feed-forward neural network (FFNN) architecture to tackle the challenging task of genres classification. The FFNN is a foundational deep learning model known for its simplicity and effectiveness in handling various classification tasks.

4.2.1 Architecture and Layers

Input Layer: The FFNN begins with an input layer consisting of 300 neurons, which accepts the Word2Vec vector representations of the textual data.

Hidden Layers: The first hidden layer comprises 500 neurons with a hyperbolic tangent (tanh) activation function. This layer is followed by a dropout layer of 0.2, promoting regularization and preventing overfitting. Subsequently, another hidden layer with 250 neurons and a tanh activation function is added, along with a dropout layer of 0.2.

Output Layer: The FFNN concludes with an output layer featuring a softmax function tailored to the specific number of classes, in our case, 6 classes (Pop, Rap, Rock, R&B, Country, and Others). The softmax function is instrumental in transforming the output into a probability distribution across the different music genres, enabling the model to provide the likelihood of each genre for a given input.

4.2.2 Training and Optimization

During the training phase, the FFNN is optimized using backpropagation and gradient descent, to minimize the loss function and enhance the model's predictive capabilities. The model undergoes iterative training processes, adjusting the network's weights and biases to improve its ability to accurately classify songs into different genres based solely on their lyrical content.

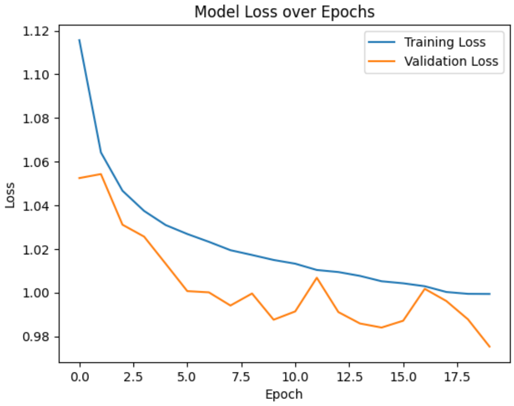

Figure 1: Loss History of the FFNN training

During the training phase, our best performing model demonstrates efficient convergence with relatively low computational requirements. Leveraging a portable computer equipped with an Intel i7-12th gen processor, we achieved notable results with training times ranging from 7 to 12 minutes ($\sim$10 epochs), all without the utilization of a GPU. In our pursuit of optimal performance, we extensively experimented with various neural network architectures and activation functions, including ReLU, sigmoid, and leaky ReLU. However, our investigations consistently revealed that the tanh activation function outperformed the alternatives for our specific problem domain. Furthermore, we observed that increasing the number of neurons beyond the current configuration did not yield a significant improvement in classification accuracy but notably extended the training time, prompting us to maintain our current architecture.

5 Experiment and Results

5.1 Experiment Setup

The experiment was conducted on a standard laptop equipped with an Intel i7-12th generation processor, operating without a GPU. The software environment included Python 3.8 and various essential libraries, such as TensorFlow, Keras, and NumPy, for neural network implementation and analysis.

For the training process, we utilized the Adam optimizer with a learning rate of 0.001. We employed a batch size of 32 for efficient model convergence. The training phase was set to run for 30 epochs, ensuring that the model had sufficient exposure to the dataset while preventing overfitting. We employed early stopping techniques and model checkpoints to mitigate the risk of overfitting and preserve the best-performing model.

Furthermore, the dataset was divided into a training set, a validation set, and a test set, with an 80-10-10 split ratio. We ensured a representative distribution of classes across the training, validation, and test sets, to reduce the risk of biased model training.

We carefully monitored the training and validation loss curves to assess the convergence and generalization capabilities of the model. Through regular monitoring, we ensured that the model was learning the essential patterns and features from the data without exhibiting signs of overfitting or underfitting.

5.2 Evaluation Process

To comprehensively assess the model's performance, we utilized a range of well-established metrics, including accuracy, precision, recall, and F1-score. The accuracy metric provided an overall understanding of the model's ability to correctly classify instances across all classes. Precision helped gauge the proportion of true positive predictions for each class, emphasizing the model's precision in classifying a particular genre. Recall indicated the model's capability to identify all relevant instances for a given class, measuring the completeness of the classification. F1-score, the harmonic mean of precision and recall, served as a balanced indicator of the model's overall performance for each class.

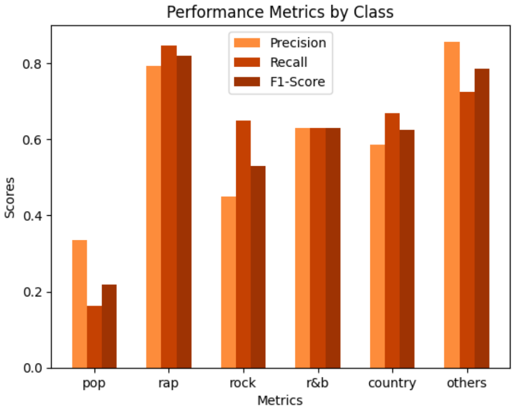

Figure 2: Performances Metric for each Genre Class

Moreover, we extensively employed a confusion matrix, a visual representation of the model's classification performance, highlighting the number of true positives, true negatives, false positives, and false negatives for each class. This analysis facilitated a granular understanding of the model's specific challenges in differentiating certain genres and provided valuable insights into potential sources of misclassification. By meticulously examining the confusion matrix, we identified patterns of misclassification and inherent complexities in distinguishing certain music genres solely based on their lyrical content.

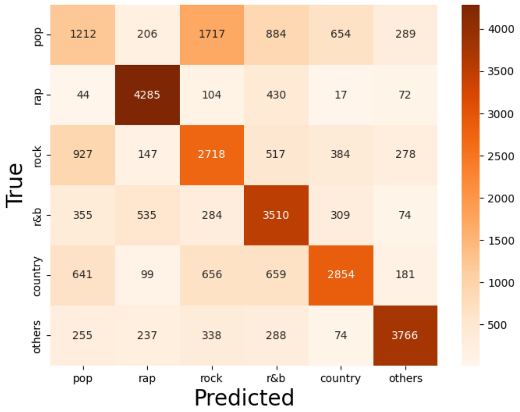

Figure 3: Confusion Matrix

5.3 Results

As detailed in the previous section we performed a rigorous experiment, in this section we will present the results that the final classification model obtained.

5.3.1 Overall Performance

The overall **accuracy** of our music genre classification model was **61.8%**. This percentage represents the likelihood that given a song the model has never seen it would be able to classify it correctly. However, as we could see in figure 2, the performance is dependent on the class, which is why it's more interesting to look at the performance per class.

5.3.2 Genre-Specific Performance

The model exhibits different levels of performance depending on the genre. Most of the metrics hovered around the 60% mark, but two notable exceptions to look at are Pop and Rap. Pop posed significant challenges with a recall of only 16.3% indicating that the model had difficulties to identify a song as being pop when presented it's lyrics. This is in stark contrast with Rap where the recall is 84.8%. In the discussion section [6] we will give an hypothesis to explain these differences.

5.3.3 Training Convergence

The early stopping mechanism that was implemented stopped the training after 18 epochs. Looking at the Loss over the Epochs in figure 1 and the results obtained, it seems that the stopping was well timed as there are no signs that of overfitting.

5.3.4 Confusion Matrix Analysis

Analysing the Confusion Matrix 3 provides more details about the misclassifications. As mentioned before Rap seems to be very distinctive and is well classified, but Pop music is often mistaken for Rock or even R&B or Country music.

6 Discussion and Conclusion

In this section we will discuss the results achieved. We will also look at the limitations of the model and suggestions for improvements and give a brief overview of the ethical implications and societal impact of this NLP solution.

6.1 Project Assessment

During this research project we successfully managed to create a method capable of classifying songs in six different genres based only on the lyrics with an overall accuracy of 61.8%. Based on this accuracy we can determine that it is indeed possible to classify songs based only on the lyrics into different categories. However the results obtained reveal certain limitations to the method.

6.2 Method Limitations

One clear limitation was found when classifying Pop songs. Pop songs are inherently diverse, they will often be a blend of multiple genres, sometimes akin towards Rock, whilst it could just as well be a Country or R&B type of song. The classification of Pop songs limited the overall accuracy one can achieve greatly. This method is also limited to English songs only as our models were only trained on English lyrics.

6.3 Dataset Issues

Even though are dataset was relatively large (≈3.5M English song lyrics), the vast majority of them were Pop or Rap songs. This meant that to have a balanced dataset to train our model on we had to discard a large amount of songs. A dataset with more entries for the lesser known genres would help identify more genres and could possibly improve the accuracy on the currently chosen genres. A second problem were the annotations present in the lyrics. Regex was used to remove most of them, but some manual verification found that not all annotations were gone. Writing more comprehensive regex could improve the cleanliness of the data.

6.4 Alternative Approaches

Within the realm of NLP using more advanced pretrained models like BERT might help in classifying Pop songs. Furthermore, it could be worthwhile to take a broader approach by incorporating audio features in addition to the lyrics.

6.5 Ethical Implications

Ethical concerns related to potential biases may arise from our genre classification models. To address this we used data sets that were as balanced as possible and would cover a great variety of music genres. Additionally, issues regarding the intellectual property and copyright aspects of the lyrics should be considered when using the dataset.

6.6 Conclusion

In conclusion, our research project resulted in a music genre classification model that demonstrates the possibility for classifying songs based solely on their lyrics. It gives insight in the way lyrics are linked to music genres, and how different genres are related to each other. The model still has limitations which may be minimized by further optimizing the data and model used, but for certain genres a more diverse approach must be taken to classify them, especially for a genre like Pop which reflects multiple genres.

References

1. Boonyanit, A., Dahl, A., & Leszczynski, M. (Year). Music Genre Classification using Song Lyrics [Stanford CS224N Custom Project]. Stanford University.

2. Google. (2013). Google News Word2Vec Embeddings. Retrieved from https://code.google.com/archive/p/word2vec/

3. JIMENEZ, M., & COSTE, S. (2023). Music Genre Classification [GitHub repository]. https://github.com/Marti2405/Music-Genre-Classification

4. Kumar, A., Rajpal, A., & Rathore, D. (2018). Genre Classification using Word Embeddings and Deep Learning. Paper presented at the 2018 International Conference on Advances in Computing, Communications and Informatics (ICACCI), Bangalore, India, pp. 2142-2146. DOI: 10.1109/ICACCI.2018.8554816.

5. Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781.

6. NIKHIL NAYAK, "Genius Song Lyrics Dataset," 2022, Kaggle, https://www.kaggle.com/datasets/carlosgdcj/genius-song-lyrics-with-language-information.