

Music genre classification based on song lyrics - comparison between different word embedding techniques and classifiers

Proof of Concept report for NLP Course, Winter 2023

Bartłomiej Eljasiak

Warsaw University of Technology
bartlomiej.eljasiak.stud@pw.edu.pl

Aleksandra Nawrocka

Warsaw University of Technology
aleksandra.nawrocka.stud@pw.edu.pl

Dominika Umiastowska

Warsaw University of Technology
dominika.umiastowska.stud@pw.edu.pl

supervisor: Anna Wróblewska

Warsaw University of Technology
anna.wroblewska1@pw.edu.pl

Abstract

Music genre classification (MGC), although a well-known task, still remains challenging in the domain of Music Information Retrieval. We tackle the problem of MGC based solely on lyrics and try to solve it using a solution composed of a state-of-the-art word embedding method tuned for this problem and a separate classification model. Our main contribution is the comparison between different word embedding methods and classification techniques, which in the domain of MGC is currently lacking. The novelty comes with an additional approach in the form of testing the impact of enriching the words with their sentiment obtained using a separate model.

1 Dataset creation

We have tried to create our own dataset using Spotify API [4] and Genius API [1]. To do that we have utilized spotipy [2] and lyricsgenius [3] libraries.

The process of data gathering is as follows. Firstly, we get recommendations from Spotify providing also the desired genre. Then, having the artist of the song and the title we obtain the lyrics from Genius. If the lyrics cannot be found, we omit the song. Lastly, we only leave the genres and the lyrics in the dataset.

This way we tried to create a dataset containing 5 genres each with 5,000 observations. Unfortunately, not all lyrics have been found. What is more, after duplicates dropping, it turned out that less than 2,000 observations were left. That means that the recommendations from Spotify repeat a lot. At the moment, we do not know if we

can influence the recommendations in some way to avoid such situations.

As the mentioned attempt was not very successful, we decided to create just a small dataset containing 4 genres (*alternative, hip-hop, pop, rock*) with 200 observations each. The final dataset contains 628 observations and was preprocessed in the same way that the other two datasets that we tested on earlier.

Unfortunately, the quality of found lyrics is also not the greatest as there exist some observations for which lyrics have lists of some artists and their songs' titles. The exemplary beginning of such lyrics is as follows:

Ring My Bell - Anita Ward
Brianstorm - Arctic Monkeys
It's No Crime - Babyface
Shooting Stars - Bag Raiders
Hellfire - Barns Courtney
I.F.L.Y. - Bazzi
Come Together - The Beatles
While My Guitar Gently Weeps - The Beatles
bellyache - Billie Eilish
my boy - Billie Eilish
ocean eyes - Billie Eilish
bad guy - Billie Eilish
bury a friend - Billie Eilish
everything i wanted - Billie Eilish
my strange addiction - Billie Eilish
when the party's over - Billie Eilish
xanny - Billie Eilish
you should see me in a crown - Billie Eilish
Uptown Girl - Billy Joel
Drowning - A Boogie Wit Da Hoodie
Sunshine Of Your Love - Cream
Get You - Daniel Caesar
Blue Day - Darker My Love...

Nonetheless, we will probably test some of the methods on this small dataset.

2 2-step CNN classifier

Since our main dataset is highly unbalanced and the genre 'Rock' has double the amount of all other samples we decided to create a new model, which will include two CNN submodels: one differentiating between the 'Rock' genre and other genres there, and the second classifying all other genres.

The first model is a CNN prepared strictly for binary classification with two sets of layers consisting of the convolutional layer, the max-pooling, and the dropout layer. As for the last classifying dense layer, we use the sigmoid activation function. While learning we have used binary cross-entropy loss function and adam optimizer.

The second model consists of the same sets of layers but repeated three times, not two. The last classifying dense layer uses the softmax activation function. While learning we have used sparse categorical cross-entropy loss function and adam optimizer.

We compared the results of the above new 2-step CNN classifier with the standard CNN classifier. For both methods GloVe embeddings were utilized, We received the following results:

Classifier	Accuracy	Bal. acc.	F1-score
CNN	54.12%	49.22%	54.02%
2-step CNN	53.30%	50.90%	53.91%

Table 1: Comparison of models

3 Sentiment

The main idea for improving our results outside improving and fine-tuning previously tested models is to combine lyric embeddings with the sentiment of the song's lyrics. This can be viewed as adding feature extraction to the current solution. We believe that providing this information straight to the classification function alongside embeddings will noticeably improve the results.

To test this hypothesis, we combined the pretrained DistilBERT model with pretrained BERT-tiny-emotion-intent. We were forced to use a small sentiment model due to hardware limitations, if those were not present, obviously, a more advanced model could provide additional improvements. This network was later fine-tuned with the sentiment model frozen. The training was done on `smallbalanced` dataset, and two networks were compared. First

sole DistilBERT and the second, the combined model, also DistilBERT with the same parameters, but additionally with the sentiment of the lyrics passed to the classification function.

We managed to slightly improve the score in the case of the second model, as can be seen in table 2. The change is not huge but, in our opinion significant enough to explore this solution further.

Model	Accuracy	Bal. acc.	F1-score
DistilBERT	62.85%	62.85%	65.18%
combined	65.07%	65.07%	67.41%

Table 2: Comparison of models

4 Authors' contribution

Self-assessed authors' contribution is in the Table 3.

Author	Contributions	Workload
Bartłomiej Eljasiak	Sentiment	33.3%
Aleksandra Nawrocka	Dataset creation	33.4%
Dominika Umiastowska	2-step CNN classifier	33.3%

Table 3: Authors' contribution

References

- [1] Genius. *Genius API*. URL: <https://docs.genius.com/> (visited on 01/04/2023).
- [2] Paul Lamere. *Spotipy*. URL: <https://spotipy.readthedocs.io/en/2.22.0/#> (visited on 01/04/2023).
- [3] John W. Miller. *LyricsGenius*. URL: <https://lyricsgenius.readthedocs.io/en/master/> (visited on 01/04/2023).
- [4] Spotify. *Spotify Web API*. URL: <https://developer.spotify.com/documentation/web-api/> (visited on 01/04/2023).