TDDE16 – Lyrics-based Music Genre Classification

Lisa Spahn Lundgren
LiU ID: lissp373
lissp373@student.liu.se

Abstract

People are constantly consuming and exploring music, and music streaming platforms such as Spotify have become indispensable in today's digital society. In order to organize the large amounts of music available on these platforms, it is essential to annotate songs with additional information such as song genre. This project explores the feasibility and performance of music genre classification based on song lyrics using NLP approaches. Eight genres are predicted: Blues, Country, Disco, Hip-hop, Metal, Pop, Reggae, and Rock. Several methods are applied, namely BoW vectorization in combination with traditional machine learning algorithms, as well as a long short-term memory (LSTM) network using pre-trained word embeddings. Lastly, an approach combining text and audio features is implemented. The experiments show that BoW vectorization in combination with a Random Forest Classifier produces the best results with an accuracy of 64.7%.

1 Introduction

In today's digital world, music streaming has become a daily activity in people's lives thanks to platforms such as Spotify, Apple Music or Soundcloud. It is estimated that approximately 100 000 songs are uploaded to these platforms every day[1]. In order to sort these large amounts of content, it is desirable to store metadata such as genre or style along with the soundtrack. This information can also be used by song recommendation algorithms, which are commonly by applications such as Spotify to engage and entertain their users (Elbir et al., 2018).

Looking at the amount of songs that are uploaded every day, it is clear that manually labeling new content is a highly labor- and time-consuming task, as it would require people listening to each song in order to determine its genre. Thus, it would be extremely useful to automatically and accurately classify music genres. Machine learning is often used to solve this task. Popular approaches are detecting genre based on the spectral features of songs, based on song lyrics, or using images such as album artwork or spectrograms (Kumar et al., 2018).

This project implements and evaluates several lyrics-based approaches, as well as combining text input (i.e. lyrics) with numerical spectral features. The aim is to compare the different approaches and understand limitations, rather than produce the optimal model.

2 Theory

Reviewing literature on lyrics-based genre classification revealed that two types of approaches are typically applied:

1. Creating a Bag of Words (BoW) representation of the text, and feeding this into a machine learning model such as Naive Bayes, Random Forest, Support Vector Machine (SVM), or Gradient Boosting (XGBoost) classifiers (Kumar et al., 2018). The BoW representation creates a vocabulary based on all words found in the training data, and creates numerical text representations by counting word occurrences. Depending on the vocabulary size, the resulting text representation is usually a sparse matrix with many zeros.

2. Word embeddings are used to avoid sparse text-corpus matrices. Words are mapped to a vector space with typically around 100 to 300 dimensions, where vectors of words with similar meaning are closer together. Neural networks are commonly used to learn this mapping, and there are several pre-trained embeddings available such as GloVe or word2vec. One can either use these embeddings directly

[1] https://variety.com/2022/music/news/new-songs-100000-being-released-every-day-dsps- 1235395788/. Last accessed: 15 Jan 2022.

with traditional machine learning models such as the ones mentioned above, or fine-tune them for a specific task using deep learning (Tsaptsinos, 2017).

3 Data

It is no trivial task to select data for genre classification. The first obstacle is defining which genres to predict. The literature review showed that this varies between publications, for instance Tsaptsinos (2017) defined five possible genres (Rock, Pop, Alternative, Country and Hip-Hop/Rap (HHR)), whereas Kumar et al. (2018) decided on the four genres Christian, Metal, Country and Rap, and Elbir et al. (2018) used ten different genres.

In a 2014 study, Sturm (2014) gives an overview of commonly used music genre data sets. It is shown that in over 50% of publications, authors create a private data set specifically for their research which is not made publicly available. The most common public data set is the GTZAN data (Tzanetakis and Cook, 2002), which was used in 23% of publications and consists of ten different genres (Blues, Classical, Country, Disco, Hip-hop, Jazz, Metal, Pop, Reggae, Rock) and 100 songs per genre, resulting in a total of 1000 songs. The first 30 seconds of each song are included as audio files, as well as a selection of spectral features over the first 30 seconds, and its spectrogram.

In order to be able to compare the results of this study to other publications, the GTZAN data was chosen for this project. Unfortunately, the data set does not include any lyrics, nor the artist and title of the songs, only the audio files, numerical and image features explained above. A GitHub user manually detected artist and title for each song and uploaded the results to a public repository[2]. Using this list, I was able to scrape the lyrics for the GTZAN data from the Genius[3] and Musixmatch[4] websites using the respective API service.

The lyrics scraping results are shown in table 1. One can see that the search did not return any lyrics for 82 jazz songs and 92 classical songs, most likely because these were instrumental tracks. Thus it was decided to exclude the Jazz and Classical genres, as they do not qualify for a lyrics-based classification approach. This leaves a total of 763 songs with lyrics for eight genres.

Genre	# songs with lyrics
Pop	100
Rock	100
Disco	99
Hip-hop	98
Country	96
Metal	96
Reggae	96
Blues	78
Jazz	18
Classical	8

Table 1: Nr. of songs for which lyrics were found per genre.

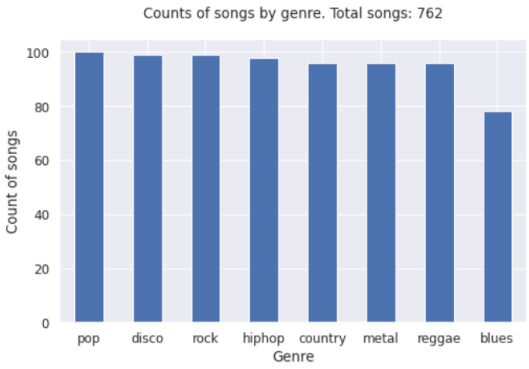

Figure 1: Data distribution by genre.

Next, the Python library `spacy` was used to detect any non-English songs. 762 songs are left after excluding these and the genre distribution can be seen in figure 1.

Lastly, some standard text processing steps were applied to the lyrics. Stop words, numbers and special characters were removed, all words were converted to lowercase and transformed to their base form (lemmatized), in order to reduce "unnecessary" words that do not contribute to the essential meaning of the lyrics, and to minimize the vocabulary. After the pre-processing, the length of the lyrics were calculated for each song and the distribution per genre is shown in figure 2. An interesting observation is that songs in the hip-hop genre tend to have much longer lyrics than any other genre. Moreover, wordclouds were generated for each genre using the pre-processed lyrics. The wordclouds can be found in the appendix figures 6 to 13.

The final pre-processed lyrics were then split

[2] https://github.com/N11K6/Song_Recommender/blob/main/dataframes/names_dataframe.csv. Last accessed: 15 Jan 2022.

[3] https://genius.com/. Last accessed: 15 Jan 2022.

[4] https://www.musixmatch.com/. Last accessed: 15 Jan 2022.

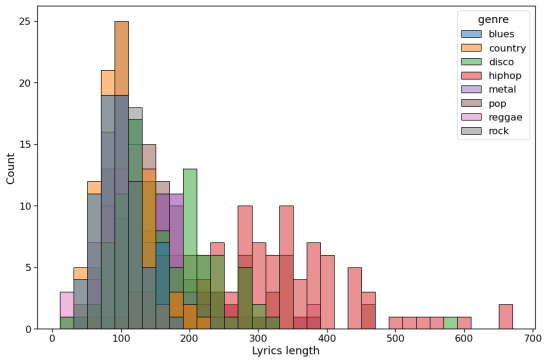

Figure 2: Number of words per song by genre.

into a training and testing set, where 80% of the data is used for training the models (602 samples) and the remaining 20% is reserved for model evaluation (153 samples). The splits were stratified according to genre, ensuring that the train and test split have the same distribution as the whole data set.

4 Method

All classification approaches were implemented using Python's `sklearn`[5] and TensorFlow's `keras`[6] libraries. A fixed seed was used for all random computations.

4.1 Traditional Machine Learning

As indicated in Section 2, there are two main types of approaches in lyrics-based genre classification: firstly using BoW representations and secondly using word embeddings. Two different BoW methods were tested: using absolute term frequencies (CountVectorizer), and using term frequency–inverse document frequency (TfidfVectorizer). For the word embeddings, GloVe[7] word representations were used, which are pre-trained on Gigaword5 and the entire English Wikipedia corpus. For each song's lyrics, the word-wise embeddings were calculated and either summed up or averaged.

The following models were used with the vectorizers: Multinomial Naive Bayes (MultNB), Gaussian Naive Bayes (GaussNB), Random Forest Classifier (RF), Support Vector Machine Classification (SVC), and Gradient Boosting Classification (XGB).

Different combinations of vectorizers and classification algorithms were tested. The following is an overview of the conducted experiments:

1. BoW Vectorizer (Tfidf/Count) + MultNB,
2. BoW Vectorizer (Tfidf/Count) + RF,
3. BoW Vectorizer (Tfidf/Count) + SVC,
4. BoW Vectorizer (Tfidf/Count) + XGB,
5. GloVe (100/300 dim, mean/sum) + GaussNb,
6. GloVe (100/300 dim, mean/sum) + RF,
7. GloVe (100/300 dim, mean/sum) + SVC,
8. GloVe (100/300 dim, mean/sum) + XGB.

4.2 Deep Learning

For the second approach, a bi-directional LSTM network (long short-term memory network) was implemented, which is a type of recurrent neural network (RNN) and has been shown to produce good results in genre classification (Tsaptsinos, 2017). LSTMs take word order into account, thereby encoding context which often leads to very high performance. The input to the LSTM network is created as follows: For each song, the embedding vector of each word is concatenated. Figure 2 showed that most genres do not have songs with more than 300 words, except for hip-hop lyrics. Thus, the maximum input length is set to 300 words, to avoid too long input matrices.

This embedding layer is then followed by two bidirectional LSTM layers, a dropout layer to prevent over-fitting, a fully connected layer with 128 hidden units and rectified linear activation function, and finally a fully connected layer with softmax activation function to create the classification output.

Two versions of this LSTM network were trained:

1. Using 100-dimensional GloVe embeddings,
2. Using 300-dimensional GloVe embeddings.

4.3 Combination of lyrical and spectral features

Lastly, a few experiments were conducted trying to combine text-based lyrics features with numerical frequency-based features. The following experiments were conducted:

[5] https://scikit-learn.org/stable/modules/classes.html. Last accessed: 15 Jan 2022.

[6] https://www.tensorflow.org/api_docs/python/tf/keras. Last accessed: 15 Jan 2022.

[7] https://nlp.stanford.edu/projects/glove/. Last accessed: 15 Jan 2022.

1. Machine learning: Concatenation of Tfidf-vectorized lyrics and spectral features. This concatenation was passed into RF, SVC and XGB classifiers.

2. Deep learning: Concatenation of word embeddings (LSTM layer output) and spectral features. This concatenation was passed into the fully connected layer.

4.4 Evaluation metrics

Mean accuracy was chosen as the evaluation metric, i.e. the average of the accuracies of each genre. This is the most common evaluation metric in music genre classification according to Sturm (2014).

For the traditional machine learning approaches, cross-validation over a grid of parameters was used to find the optimal hyperparameters (vectorizer parameters and model parameters). For the deep learning approach, I trained the network multiple times with different training parameters such as number of epochs and learning rate, and selected the settings with the best accuracy results.

5 Results

Model	Params	Accuracy
BoW + MultNB	Tfidf	53.6 %
BoW + RF	Count	**64.7** %
BoW + SVC	Tfidf	51.6 %
BoW + XGB	Tfidf	57.5 %
GloVe + GaussNb	300d, mean	36.6 %
GloVe + RF	100d, sum	49.7 %
GloVe + SVC	300d, sum	49.7 %
GloVe + XGB	100d, sum	49.0 %
LSTM	100d	**49.7** %
LSTM	300d	48.4 %
Concat + RF	Tfidf	51.0 %
Concat + SVC	Tfidf	45.8 %
Concat + XGB	Count	**53.6** %
LSTM Concat	100d	45.8 %

Table 2: Model performances.

The results of all experiments are shown in table 2. One can see that the best performance overall was achieved using the BoW CountVectorizer in combination with a Random Forest Classifier, achieving 64.7% accuracy. The confusion matrix is shown in figure 3.

The best performing LSTM setup was using 100-dimensional word embedding vectors with 49.7% accuracy. The confusion matrix and training loss are shown in figure 4.

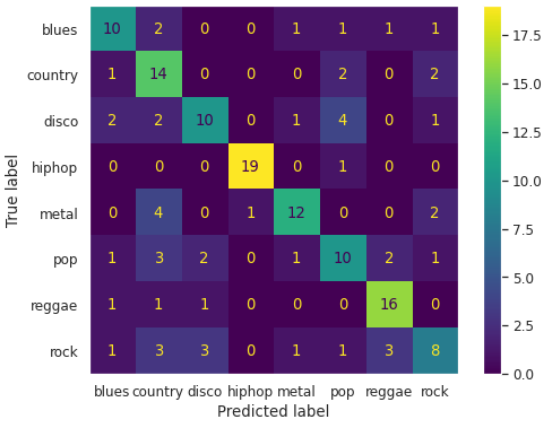

Figure 3: Confusion matrix of best model overall (64.7% accuracy): CountVectorizer and Random Forest Classifier.

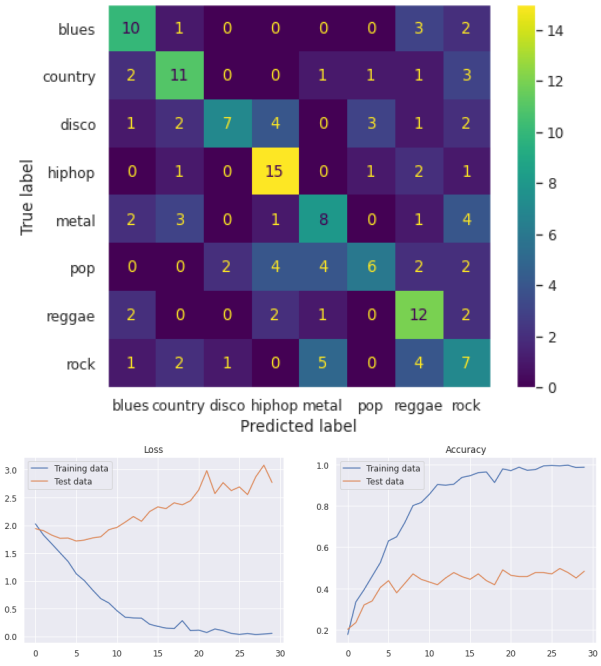

Figure 4: Confusion matrix and training loss of best LSTM model (49.7% accuracy), using 100-dimensional word embeddings.

The best performing combined approach, which was also the 3rd best approach overall, was the concatenation of Count-vectorized lyrics and spectral features with Gradient Boosting Classifier, achieving 53.6% accuracy. The confusion matrix is shown in figure 5.

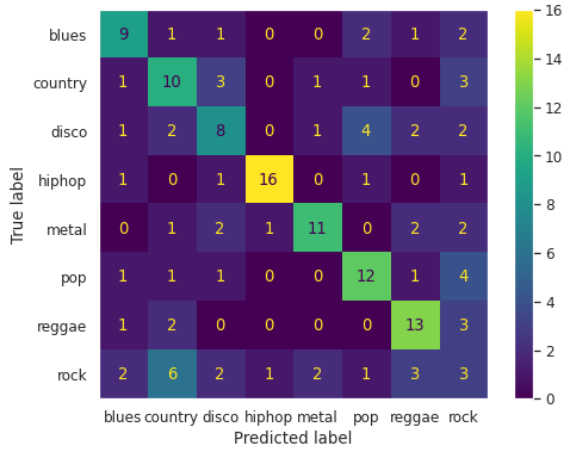

Figure 5: Confusion matrix of best combined model (53.6% accuracy): lyrics encoded using CountVectorizer, concatenated with spectral features. Gradient Boosting Classifier.

6 Discussion

In order to put the achieved results into perspective, an overview of performances achieved by other researchers on the GTZAN dataset can be found in table 3. Though it has to be noted that these accuracies are for the ten-genre classification problem, this project only considered eight genres.

Paper	Mean accuracy
Tzanetakis and Cook (2002)	61.0 %
Elbir et al. (2018)	66.0 %
Patil and Nemade (2017)	77.8 %
Ajoodha et al. (2015)	81.0 %
Sturm (2013)	82.5 %

Table 3: References for mean accuracy on GTZAN data.

Initially, I was very satisfied with the results of the best model reaching nearly 65% accuracy. 8-class genre classification seems like a difficult problem, as some genres, such as pop and country, or rock and metal, are not clearly distinguishable in my opinion. However, looking at the performance references in table 3, it becomes clear that other classification approaches are able to achieve much higher accuracies. Moreover, their model considers two additional classes jazz and classical. Taking the reduced number of classes into account, the best approach of this project is comparable to the baseline established by the creators of this data set Tzanetakis and Cook (2002) from over 20 years ago.

Apart from the publication that reaches 66% accuracy, all other methods use frequency-based features that are extracted from the audio signal of the song. Thus it seems that spectral features seem to be a better predictor than song lyrics, at least for the data and approaches of this project. This inspired the idea to concatenate spectral and text data, in order to utilize both feature sets. Surprisingly, the models using the combined feature sets performed worse than the purely lyrics-based approach. A possible explanation is that using both feature sets increased the input data's dimensionality and complexity, though the amount of training samples remained the same. Therefore, the models might require more data in order to better learn the input-genre relationship.

The same applies to the LSTM approaches. In literature, RNNs such as the models implemented here consistently outperform traditional machine models. Though in this project, the opposite scenario occurred. Possibly, the reason for this is again the lack of data. Deep learning architectures have more parameters that need to be trained as opposed to simpler algorithms that train much faster.

Another surprising result was that classifiers using the GloVe vectorizer performed worse than the same models using a BoW vectorizer. Looking at several song lyrics and the word clouds in the appendix, one possible explanation is that lyrics seem to contain an above-average amount of slang and filler words which do not carry much meaning, such as "aha", "yeah", "yo", "mmm" or "jammin'". Maybe the pre-trained GloVe corpus does not contain these words or has difficulties encoding their meaning, whereas BoW vectorizers build their corpus solely based on the training data, in this case the song lyrics, and only consider word occurrences, not word meaning.

The last result I would like to discuss is that, looking at the confusion matrices, it stands out that the genres *hiphop* and *reggae* have the highest accuracies. Looking at the wordclouds, one can see that these two genres seem to have the most distinctive lyrics (9, 12), all other wordclouds are rather similar and not easily distinguishable. The genre which is the hardest to classify is *rock*, it is often miss-classified as metal, country or reggae. From a personal perspective, I can agree that it can be hard to draw a clear line between these genres.

This leads to one major concern with genre classification, which is that music genre is subjective. There is no clear definition of how many genres

there are or which are the main genres. In this project, one of eight possible genres was predicted. However, websites such as MusicGenresList[8] indicate that there are many more genres: around 40 main genres with a total of circa 400 sub-genres. Thus, assigning genres to songs is a subjective and ambiguous decision that is difficult even for humans.

Another noteworthy point is that, while GTZAN is the most common data set for classifying music genres, it has been shown to contain errors. Sturm published a study in 2012, showing that approximately 11% of the songs are mislabeled and that 5% of the data are duplicates (Sturm, 2012). Some samples are the same song performed by different artists, meaning that the audio track is different, however the lyrics for these samples are nearly identical. This property could have affected the models in this study more than for instance audio-based approaches.

7 Conclusion

To conclude this project, I showed that lyrics-based classification is feasible, however it does not reach state-of-the-art performance compared to other approaches applied to the same data set. One reason for this might be a lack of data. Numerical representations of text data can be high-dimensional, and thus state-of-the-art NLP models such as LSTMs or Transformers require large amounts of training data. For future work, it would be interesting to see if incorporating more data would change the results, and I would allocate more time for cleaning and quality-checking the data. Especially when scraping text such as lyrics from the internet, the data collection process takes much longer than expected.

References

Ritesh Ajoodha, Richard Klein, and Benjamin Rosman. 2015. Single-labelled music genre classification using content-based features. In *2015 Pattern Recognition Association of South Africa and Robotics and Mechatronics International Conference (PRASA-RobMech)*, pages 66–71.

Ahmet Elbir, Hilmi Bilal Çam, Mehmet Emre Iyican, Berkay Öztürk, and Nizamettin Aydin. 2018. Music genre classification and recommendation by using machine learning techniques. In *2018 Innovations in Intelligent Systems and Applications Conference (ASYU)*, pages 1–5.

Akshi Kumar, Arjun Rajpal, and Dushyant Rathore. 2018. Genre classification using feature extraction and deep learning techniques. In *2018 10th International Conference on Knowledge and Systems Engineering (KSE)*, pages 175–180.

Nilesh M Patil and Milind U Nemade. 2017. Music genre classification using mfcc, k-nn and svm classifier. *International Journal of Computer Engineering In Research Trends*, 4(2):43–47.

Bob L. Sturm. 2012. An analysis of the gtzan music genre dataset. In *Proceedings of the Second International ACM Workshop on Music Information Retrieval with User-Centered and Multimodal Strategies*, MIRUM '12, page 7–12, New York, NY, USA. Association for Computing Machinery.

Bob L Sturm. 2013. On music genre classification via compressive sampling. In *2013 IEEE International Conference on Multimedia and Expo (ICME)*, pages 1–6. IEEE.

Bob L. Sturm. 2014. A survey of evaluation in music genre recognition. In *Adaptive Multimedia Retrieval: Semantics, Context, and Adaptation*, pages 29–66, Cham. Springer International Publishing.

Alexandros Tsaptsinos. 2017. Lyrics-based music genre classification using a hierarchical attention network. In *International Society for Music Information Retrieval Conference*.

George Tzanetakis and Perry Cook. 2002. Musical genre classification of audio signals. *IEEE Transactions on Speech and Audio Processing*, 10:293 – 302.

A Lyrics wordclouds

Figure 6: Blues lyrics wordcloud.

[8] https://www.musicgenreslist.com/#music-genre-list. Last accessed: 15 Jan 2022.

Figure 7: Country lyrics wordcloud.

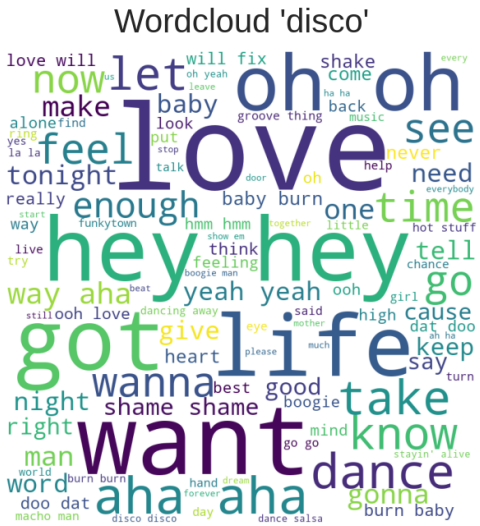

Figure 8: Disco lyrics wordcloud.

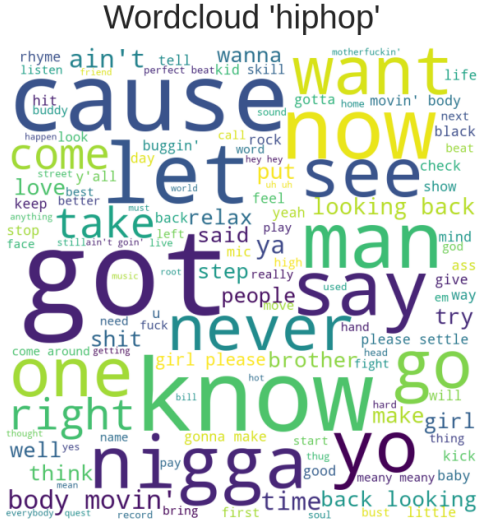

Figure 9: Hip-hop lyrics wordcloud.

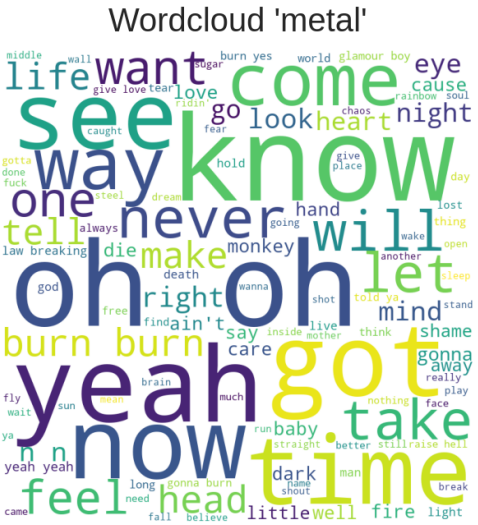

Figure 10: Metal lyrics wordcloud.

Figure 11: Pop lyrics wordcloud.

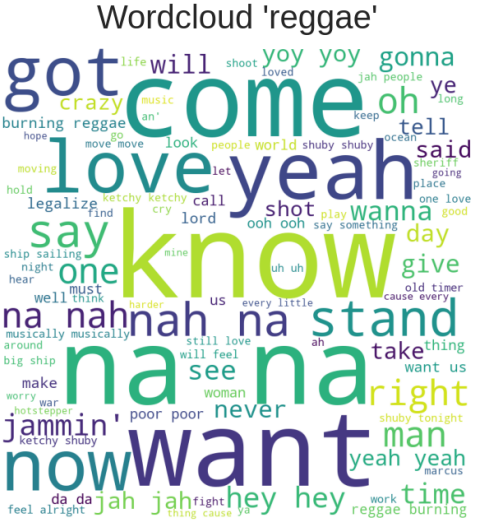

Figure 12: Reggae lyrics wordcloud.

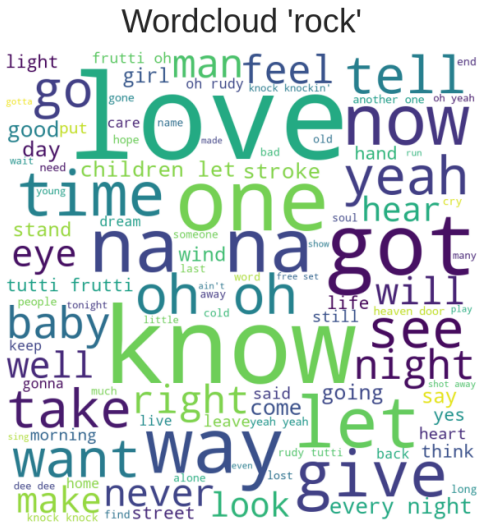

Figure 13: Rock lyrics wordcloud.