Classification of Music Genres with Logistic Regression Based on Lyrical Content

Anton Philipp Paul
Projektseminar
31/03/2020
Matr. No.: 2925824

Abstract

Genre classification of songs is mostly based on auditory features of specific genres and not as much on the lyrical content of songs. By looking closer at the data provided in song lyrics, one can construct a method of classifying the song's genre by only taking into account the lyrics. The resulting task is to build different genre classifiers which work on the basis of song lyrics which have been analysed with the tf-idf method and figure out the genre with the most unique lyrics, thus leading to the most accurate classifier.

1 Introduction

When trying to classify music genres, the focus mostly lies on auditory features like loudness, timbre and music segments. The lyrics themselves, however, already carry a lot of information about the song's genre in the frequency of certain vocabulary. The question now is which genre offers the most unique set of words based on which the classifier with the highest accuracy can be constructed.

In the first section, I will present the dataset which is used in this analysis and describe the preparation process. In the data analysis section, I take a look at some characteristics of each genre, which already provide informative data. These characteristics being the type/token ratio and the affective rating of adjectives, nouns and verbs. Then, the preprocessed data undergoes some last preparation steps before one classifier per genre is trained and tested on said data. After that, the results are evaluated and interpreted and the analysis is concluded in the last section.

2 Preprocessing of Data

The dataset used for this analysis is the ”55000+ Song Lyrics” dataset from kaggle (Kuznetsov, 2017), which was scraped from LyricsFreak. It consists of a variety of artists and songs all in English and also includes a link for each song, which is intended to be combined with the LyricsFreak website name for possible lookups. Since this link was never used in this project, it will not be included in any of the data excerpts or analyses. An excerpt of this dataset would then look like the following:

Artist	**Song**	**Text**
ABBA	You Owe Me One	Lyrics
Ace of Base	Always Have, Always Will	Lyrics
Ace of Base	Cecilia	Lyrics

Table 1: Excerpt of original dataset

In this dataset, the genre was not yet attributed to the different artists, so I decided to use the *musicbrainz* online database (MetaBrainz) to automatically determine the respective genres and add them to the file via the *musicbrainzngs* package (Porter and et. al). There were multiple genres given for each artist with an additional counter per genre. Since musicbrainz also has a heavy community focus, up to a certain point users could submit their own genres for artists and these submissions could then be upvoted and downvoted by the community. The counter next to each genre in the database represents these community votes and throughout my research I noticed that the more obscure/niche the submitted genre was, the less upvotes it received, so I was confident that the most probable genre would always have the most upvotes. As a result of this observation, I attributed the genre with the highest count to each artist. If no genre could be found, the artist was attributed the tag ”Unknown”, so I could filter this artist out more easily in a later step.

This genre attribution, however, did not take place within the original file, as the dataset was too large to iterate over properly, so I decided to split the original file into files per artist. For each artist found in the original file, I created a new .csv file, which then contained the artist's songs and genre. This led to 643 new files and the following amount of data:

Type of Data	Amount of Data
Songs	57650
Artists	643
Genres	161

Table 2: Data of original dataset after genre attribution

With a total of 161 genres, there was a variety of genres with only a few artists, hence not a lot of data to filter out the desired information. I decided to limit the data to genres which were attributed to at least ten artists. Since this analysis is focused on the characteristics of specific genres, I restructured the data from the differentiation between artists (643 files) to the differentiation between genres (11 files); I created a text file per genre which included all song lyrics from all artists which belonged to said genre. The result was:

Type of Data	Amount of Data
Songs	34971
Artists	328
Genres	11

Table 3: Valuable data after filtering

Genre	Occurrences
Rock	89
Pop	87
Hip Hop	29
Country	19
Alternative Rock	18
Hard Rock	17
Soul	16
Folk	15
Progressive Rock	14
Jazz	13
Heavy Metal	11

Table 4: Genres with at least 10 attributed artists

As can be seen in table 4, the two genres ”Rock” and ”Pop” had a disproportionally high number of artists, which might have influenced the final results in negative ways, which is why I decided to limit the song lyrics I would analyse to a maximum of 25 artists.

Next, the data was prepared with the help of the Natural Language Toolkit (Loper and Bird, 2002). Furthermore, utterances like ”Chorus”, ”Repeat chorus”, ”Verse:”, ”Bridge”, etc. and instructions printed in brackets, e.g. ”(Guitar solo)”, had to be removed from the text files and when a line only consisted of the word ”Solo”, it should be dropped as well. Finally, this newly saved cleaned data was tokenized and saved in yet another text file, now ready for the last step.

As the last preprocessing step, the genre files were tagged with the help of the TreeTagger (Schmid, 1999).

3 Data Analysis

In order to learn more about the lyrical content of each genre and be able to make educated assumptions about potential classification results, I decided to look at the type/token ratio of each genre and, in a further step, the affective rating of different parts of speech.

3.1 Type/Token Ratio

The first step of the data analysis consisted of looking at the type/token ratio of all genres. In order to do this, the type/token ratio of every song for each genre was determined, from which the mean ratio for the respective genre was calculated.

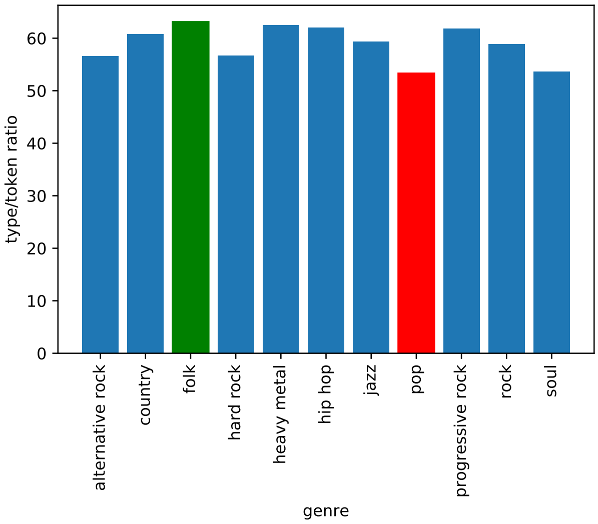


Figure 1: Type/token ratios of genres

Figure 1 shows these average type/token ratios. The highest and lowest values are Folk with 63.11 and Pop with 53.31. Other high ratios were determined for Heavy Metal, Hip Hop, Progressive Rock and lower ratios for Alternative Rock, Soul and Hard Rock.

3.2 Affective Ratings

Besides the type/token ratio, the affective rating of different parts of speech was also taken into account and analysed. All tokens of the part of speech relevant for the respective analysis were filtered out from the tagged files and then, depending on the part of speech, the average affective rating of each token was looked up in the affective rating list by Warriner, Kuperman and Brysbaert (Warriner et al., 2013). These ratings were furthermore divided into positives and negatives (with positives ranging from a value of 5 and vice versa) and the respective percentages in regard to the total number of tokens of that part of speech are returned. This analysis returned the following results:

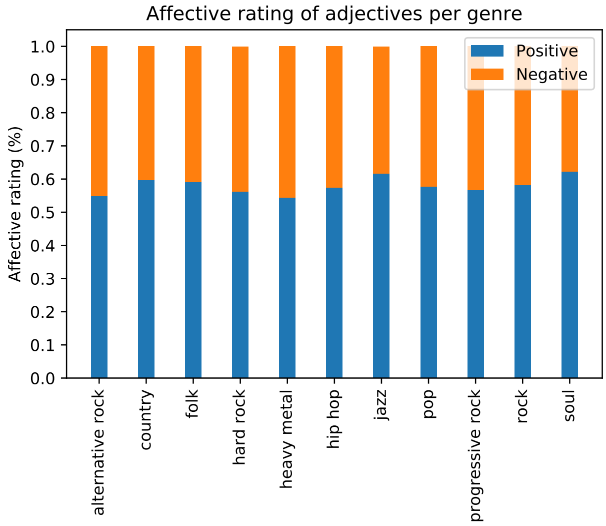


Figure 2: Affective rating of adjectives

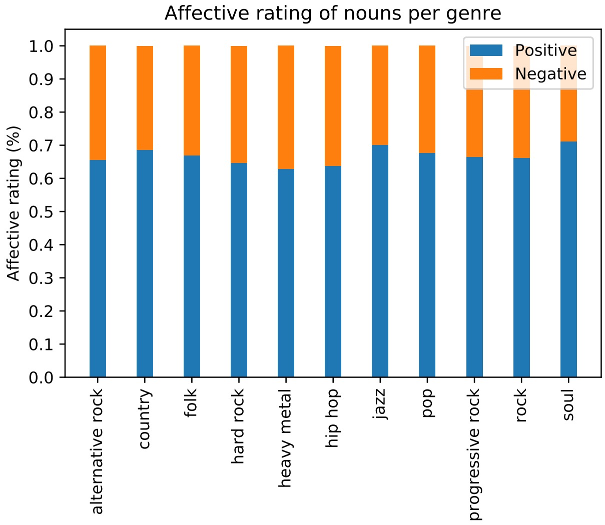


Figure 3: Affective rating of nouns

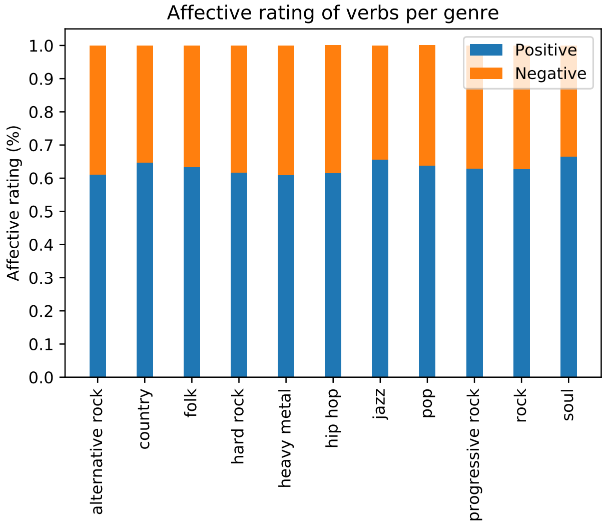


Figure 4: Affective rating of verbs

In these figures it becomes clear that Alternative Rock shows the most negative results in the affective rating of adjectives and verbs and Heavy Metal takes that position for nouns with Hard Rock, Hip Hop, Rock and Progressive Rock following closely. Soul exhibits the most positive rating in all three parts of speech with Jazz following as a close second and Pop, Folk and Country also returning more positive results.

3.3 Evaluation of Ratings and Ratio

When examining the results of the affective rating analysis, it becomes apparent that no matter the part of speech, the differences between the genres are too minimal to be considered a valid basis on which to differentiate the genres. While general trends become evident, such as Soul and Jazz exhibiting more positive affective ratings and genres such as Heavy Metal and Hip Hop returning more negative ones, even the two extremes on each scale are too close to each other to be taken as conclusive evidence that ratings in that range can be attributed to the respective genre; this would only result in a large overlap of genre attributions, which is not the goal of this project.

The type/token ratios on the other hand, while still returning certain inconclusive outcomes, show some more promising results in the gap between the two extremes. One can clearly see the difference in the type/token ratio between genres like Folk and Heavy Metal as examples with a high ratio and Pop and Soul with a low one. The lyrical content of many songs belonging to the genre of Pop is often criticized for its repetitiveness, lack of creativity and simplicity, which manifests itself in a low type/token ratio. While this is still not a conclusive genre-defining result, it shows a clear trend, which the final classifier could represent as well: The more unique a genre's vocabulary, the more successful will the classifier be in attributing the correct genre after the application of the tf-idf weighting scheme.

4 Classification

4.1 Final Data Preparation

In order for the classifier to be trained on the lyrics of multiple genres and not only the genre it should detect, I needed to reorganize the data. Up to this point, I was using one .csv file per genre, which contained all songs of said genre. Now I needed to reorganize all these files into one file containing all songs and their respective genre attribution.

The next step was to redefine certain characteristic in this file so that a logistic regression classifier could work with the data. Since this kind of classifier can only work with numerical values as classes, I could not use the actual genre names as values for each song. This led to the decision to use the genres as table columns and assign each song a 0 or 1 depending on if they belonged to said genre or not; 0 representing that the song does not belong to the genre and vice versa. The resulting dataframe looked as follows:

Song	**alternative_rock**	**country**	**folk**	**...**
Lyrics	1	0	0	...
Lyrics	0	0	1	...
Lyrics	0	1	0	...

Table 5: Example Excerpt of Songs and Genres Dataframe

The original file of the excerpt shown in table 5 had 12 columns and 21616 rows, one per song. This would then be the final file used for the actual classification.

4.2 Logistic Regression Classification

For the classification, I decided to train one logistic regression classifier per genre and evaluate and compare the results with a confusion matrix and classification report. In order for the logistic regression classifier to return valuable results, I needed to reexamine the available data. I realized that the difference in the amount of occurrences for both classes (0 and 1) was too big for the classifier to be able to gather enough information on the 1 class and classify the songs correctly, as can be seen in table 6.

Genre	**0**	**1**	**Ratio 1/0**
Alternative Rock	18032	1422	**0.08**
Country	16899	2555	**0.15**
Folk	17982	1472	**0.08**
Hard Rock	17521	1933	**0.11**
Heavy Metal	18388	1116	**0.06**
Hip Hop	17966	1488	**0.08**
Jazz	18293	1161	**0.06**
Pop	16697	2757	**0.17**
Progressive Rock	18169	1285	**0.07**
Rock	16676	2778	**0.17**
Soul	17967	1487	**0.08**

Table 6: Ratio of Classes 1 and 0

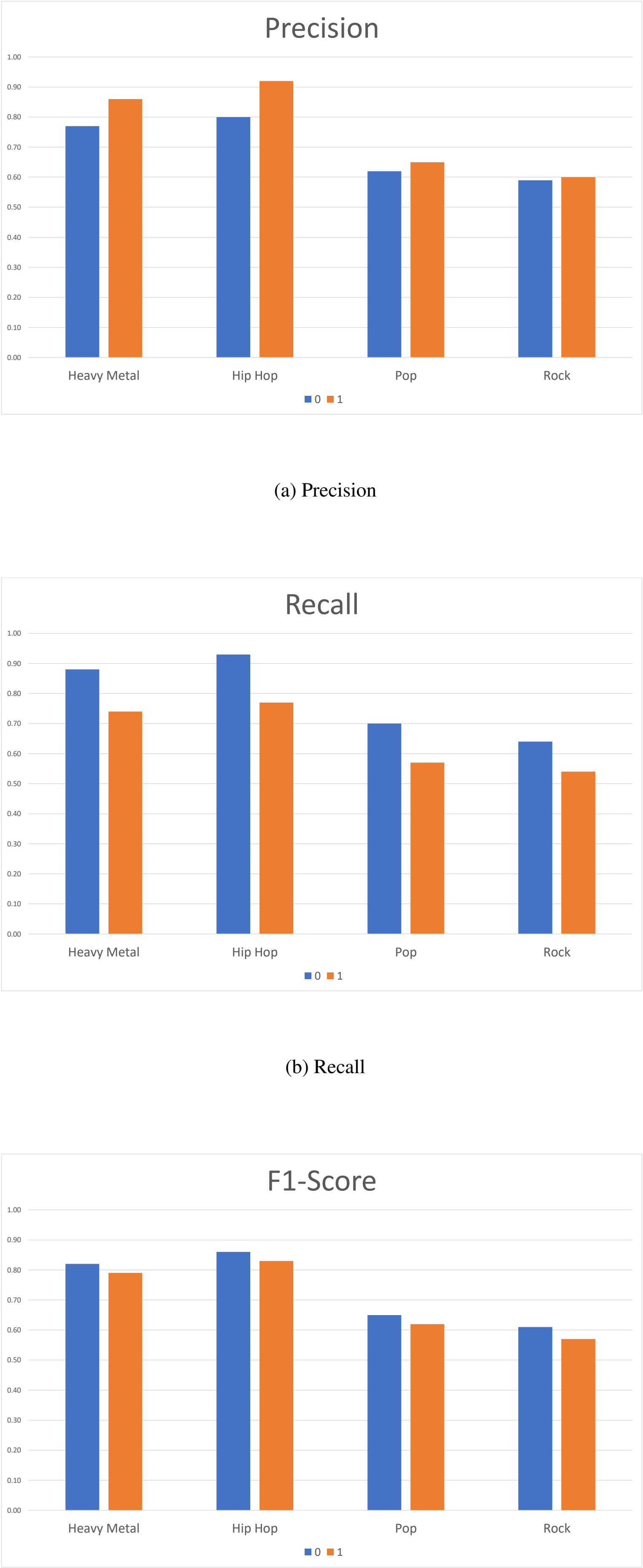
This uneven distribution would make it nearly impossible for the classifier to learn the characteristics of the songs belonging to a genre, as it would be overrun with examples for songs not belonging to said genre. Hence, I decided to use the bootstrapping method to resample the data. This method takes samples of each class and either downsizes or upsizes the classes, so that the dataframe includes an equal amount of samples for each class. In this case, I decided to limit the sample number for each class to 2000 in the training set and 200 in the testing set, since that was roughly the medium between the numbers of the occurrences of 1 for all genres. This meant that only 2000 examples of songs not belonging to a given genre would be included in the training of the classifier and if there were more than 2000 examples of a song belonging to the genre, some of these were also not included. If, however, there were less than 2000 examples of the class 1 for a genre, some of them were included more than once to upscale the class samples. It might also be important to note that for each genre, therefore for each iteration of the classifier, the bootstrapping method was applied anew, hence shuffling the data every time.

With this new dataframe containing a total of 4000 songs, the next step necessary for the classifier could be performed. Training the classifier solely on the lyrics of each song would not deliver the results I was wishing for, which is why I decided to use the tf-idf weighting method. This method first counts the term frequency (tf) for each song and ranks all words based on the amount of times it occurs in a document. However, certain words like ”go” or ”say”, which might appear across many different documents with a high frequency, do not carry enough value to be considered a characteristic for the genre it appears in. That is where inverse document frequency (idf) comes in. This lowers the importance of terms appearing in many documents and raises the importance of terms which do not appear regularly. Therefore, with this tf-idf method, terms which are unique to certain documents become more important for the classification and terms appearing throughout a number of documents are considered less valuable.

After the training and testing data was weighted with this method, the logistic regression classifier was trained on the tf-idf training data together with the binary data of class affiliation for each song in the training set. Then, predictions for the test data were made based on the trained classifier and a classification report was created. This process was performed for each genre using a function, so that a total of 11 classifiers was trained and tested in one sitting. In the next section, the classification reports will be analysed further to evaluate the significance of the results.

4.3 Classification Report

In order to gather more conclusive data, I decided to collect the classification reports containing the precision, recall and F1-score for each genre for 10 repetitions of the classification. I then calculated the mean values for the two genres with the highest results and the two with the lowest results to create a contrast which could then be further interpreted. The genres with the highest scores were Heavy Metal and Hip Hop and the ones with the lowest scores were Pop and Rock. The resulting mean diagrams can be seen in figure 5 on the following page.



(a) Precision

(b) Recall

(c) F1-Score

Figure 5: Mean Values of Classification Report

Now to take a closer look at the numbers in figure 5. Starting with the precision diagram, it first should be mentioned that the scores are higher for class 1 for each genre. Precision is the number calculated from the number of times the classifier correctly attributed the class divided by the number of times it attributed the class overall. Here Hip Hop shows the highest result with a precision of 0.92 and Rock comes last with a score of 0.6. This can, however, already be taken as a positive sign for the successfulness of lyrical content as a classifying characteristic for genres. The classifiers do not try to blindly attribute a class to each sample, which could still lead to a good score. Instead, it makes educated guesses and, in case of Hip Hop, is correct most of the time.

In the second diagram containing the recall scores, we see the results of when the classifier predicted the actual class of a song; i.e. the times it correctly guessed the class divided by the total number of occurrences of said class. Here the scores for 0 are always higher than for 1, however we still have a clear contrast between Hip Hop and Heavy Metal with scores of 0.74 and 0.77 and Pop and Rock with scores of 0.57 and 0.54 for 1. The classifier seems to be more accurate in guessing that a song is not part of a genre than when it is. This still shows a strong capability of the classifier, especially in the case of Hip Hop, where it correctly assigned 0 to a song in 93% of all occurrences of 0.

Finally, the F1-Score represents the mean of precision and recall and therefore can be interpreted as a measure of the classification's accuracy. Here the scores for 0 are once again higher than for 1, however the gap between the respective pairs are smaller. Hip Hop and Heavy Metal are still displaying the high scores with 0.79 and 0.83 and Pop and Rock the low scores with 0.62 and 0.57 for the class 1.

4.4 Interpretation of Results

When the type/token ratios of the different genres were analysed in section 3.3, the observation was made that genres like Heavy Metal and Hip Hop displayed a higher score for the ratio while Pop delivered a weaker result. This trend can now also be observed in the actual classification. Genres that are generally considered to be more repetitive find themselves with lower accuracy scores, since it was more difficult for the classifier to filter out unique characteristics for said genre in comparison to all others. Hip Hop as a prime example of a genre with a lot of variety and songs containing a higher amount of words in general is now the genre which the classifier can identify most easily. This shows that on the one hand, the lyrical content of songs can most certainly provide enough data to successfully classify certain genres simply on the basis of this characteristic, while on the other hand this same feature can be a disadvantage for other genres, which appear to be more simple in its vocabulary.

What, however, came as a surprising result, is that Rock in fact turned out to be the genre returning the poorest results with the classifier while it had a significantly higher type/token ratio than Pop. But only looking at the ratios in figure 1, we can also see that Folk and Progressive Rock should have returned the most positive results. So the type/token ratio alone could not be the deciding factor in the classification. What pushed other genres to the top and the bottom of the scale was the tf-idf weighting method. While Rock had a higher type/token ratio, the tf-idf weighting of all Rock songs might have found that a lot of its vocabulary appears in many other songs and it only offers few unique words. The same applies to Hip Hop which was pushed to the top to replace Folk based on the vaster amount of unique vocabulary in Hip Hop songs.

5 Conclusion

What I set out to do with this project was to create a classifier that would return the highest accuracy for the genre with the most unique lyrics. Looking at the results for Hip Hop, an accuracy of 83% is not a perfect score but it is a high score nonetheless, especially when contrasted with the lower scores of Rock and Pop and considering actual facts and opinions on the genres and their lyrical content. Hip Hop generally has a high word count, which increases the possibility of more unique words appearing in Hip Hop songs than in other genres, which can cleary be seen in this analysis.

For future analyses in a similar direction, I might consider finding a dataset with a more balanced distribution of the classes I want to examine. The fact that I had to downscale the songs not belonging to a genre and upscale those that do, could have affected the final results. If I had a dataset with equally distributed classes to train the classifier without having to ignore a big part of the data, the accuracy might have been even higher.

This project successfully showed that songs cannot only be classified based on auditory features but that their lyrical content alone already provides enough information to differentiate between genres and that the more unique a genre's lyrics are, the easier it is for a classifier to learn the genre's characteristic features and correctly attribute the respective class.

References

Sergey Kuznetsov. 2017. 55000+ song lyrics. https://www.kaggle.com/mousehead/songlyrics.

Edward Loper and Steven Bird. 2002. Nltk: The natural language toolkit. In *Proceedings of the ACL-02 Workshop on Effective Tools and Methodologies for Teaching Natural Language Processing and Computational Linguistics - Volume 1*, ETMTNLP '02, pages 63–70, Stroudsburg, PA, USA. Association for Computational Linguistics.

MetaBrainz. Musicbrainz. https://musicbrainz.org/. Accessed on 26-03-2020.

Alastair Porter and et. al. musicbrainzngs documentation. https://readthedocs.org/projects/python-musicbrainzngs/downloads/pdf/latest/. Accessed: 26-03-2020.

Helmut Schmid. 1999. Improvements in part-of-speech tagging with an application to german. In *Natural language processing using very large corpora*, pages 13–25. Springer.

Amy Beth Warriner, Victor Kuperman, and Marc Brysbaert. 2013. Norms of valence, arousal, and dominance for 13,915 english lemmas. *Behavior research methods*, 45(4):1191–1207.