Repetitiveness in Lyrics: an Insight in Music Genre Classification

Josip Jukić, Jeronim Matijević, Mate Mijolović

University of Zagreb, Faculty of Electrical Engineering and Computing
Unska 3, 10000 Zagreb, Croatia
{josip.jukic,jeronim.matijevic,mate.mijolovic}@fer.hr

Abstract

This work proposes two different approaches to music genre classification task based on lyrics. Firstly, model grounded on *fastText* embeddings was introduced, which ultimately delivered the best performances. As a different setup regarding this matter, we offered a model with custom crafted features, constructed to capture repetition structures in lyrics. Consensus clustering algorithm was considered as a feature selector in this process. Importance of repetitiveness in lyrics is further explored, especially in the view of genre classification. We shaped this task to utilize convolutional neural networks in lyrical pattern recognition. Performances of these models were evaluated on two different datasets: MetroLyrics and MusixMatch which contain 10 and 4 different genre types respectively. Our work tends to indicate that lyrical repetition can be exploited in genre classification, especially in *hip-hop*, *pop* and *R&B*.

1. Introduction

Managing your music libraries can be a tedious task. This is posing an increasingly hard problem nowadays because of the growth of online music databases and easy access to music content. One way to categorize and organize songs is based on the genre. Although the division of music into genres is somewhat arbitrary and subjective, a particular genre can still be described within some characteristics of the music such as rhythmic structure, harmonic content and instrumentation (Tzanetakis and Cook, 2002).

Gjerdingen and Perrott (2008) have shown that humans are remarkably good at genre classification. As a matter of fact, humans can accurately predict a musical genre based on 250 milliseconds of audio. This finding suggests that we can differentiate genres using only the musical surface without constructing any higher level theoretic descriptions. Therefore, it is not surprising that most of genre classification for digitally available music had been done manually until recently. Because of a surge in quantity of produced songs, automatic genre tagging would be beneficial for both music streaming services and users.

Most of the approaches to genre classification imply using audio features. We chose to work only with songs lyrics while tackling this problem. Our motivation lies in faster processing of text when compared to audio analysis. Additionally, audio files can be hard to procure in wanted formats and they are often substantially larger in size than lyrics as well. Intrigued by a visual essay on repetitiveness in pop lyrics (Morris, 2017), we decided to further explore the importance of repetition in song lyrics. Simple statistical analysis can show that the music is becoming more and more repetitive, especially pop songs. Interestingly enough, this trend is particularly obvious and more intense in top 10 songs on charts throughout the years.

Since it is generally much harder to classify genres using solely lyrics, our main goal was to achieve results that can rank with audio-based classifiers in which we eventually succeeded. In order to do so, we tested the performance on shallow models such as Support Vector Machine (SVM) and Logistic Regression with various feature sets. Among them was a set based on TF-IDF scheme and *fastText* library which we used for learning word embeddings. Described model was also utilized for comparison with repetitiveness feature model on which we put a special focus. Both of our models heavily outperformed the used baseline model.

We decided to implement consensus clustering to select specific features for lyrics repetition. It represents a proxy task to estimate the quality of scrutinized features. Quality of clustering using specific features was translated as their importance. Our hypothesis is that the general song clustering task is considerably correlated with genre classification problem, therefore we use it as a feature selector in this fashion. We hope to achieve a more robust estimation since we can use unlabeled data without restricting ourselves to a specific dataset. After constructing and selecting the features, we evaluated all variants and compared the repetition features model with audio classifier baseline.

2. Related Work

Genre classification is dominantly approached by audio analysis. However, regarding lyrics-based methods, most of them involve feature engineering. In more recent works, recurrent neural networks were applied for this task. More precisely, hierarchical attention network was adapted to song lyrics (Tsaptsinos, 2017). This is supported by a hierarchical layer structure that lyrics exhibit - words combine to form lines, lines form segments, and segments form a complete song. Another idea that inspired this project was a clustering method based on consensus. We stumbled upon it in gene expression analysis where it was used as a method for class discovery (Monti et al., 2003). It is depicted as a robust method that can validate the number of clusters. Furthermore, the method deals with the problem of random center initialization.

In our work, we combined several ideas. Consensus clustering was used as a feature selector and also as a general indicator of feature quality. Compression algorithm served to estimate repetitiveness of song lyrics. Besides that, we exploited repetition structures in order to discover a connection between lyrics and accompanying songs' audio features. In the attempt to do so, convolutional networks were utilized to capture patterns in repetition structures.

Text Analysis and Retrieval 2019: Course Project Reports (TAR 2019), pages 31–35, University of Zagreb, Faculty of Electrical Engineering and Computing, Zagreb, July 2019.

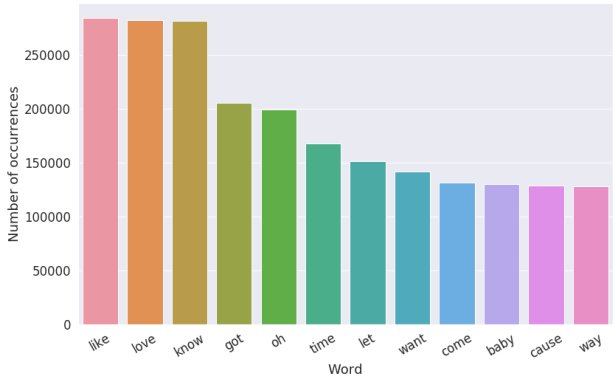

Figure 1: Word frequency on the MetroLyrics dataset

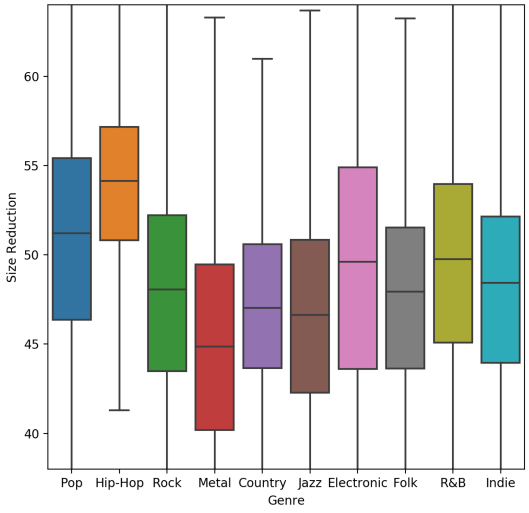

Figure 2: Boxplot of lyrics size reduction by genres

3. Datasets and Preprocessing

Since building a custom dataset was not feasible due to time restrictions, we decided to use *380.000+ lyrics from MetroLyrics* dataset from Kaggle. The dataset contains song lyrics ranging from year 1970 up to 2016, along with metadata such as *song name*, *genre* and *artist name*. The raw dataset is multilingual, although the dominant language is English. To filter out non-English content, we used statistical language detector based on Naive Bayes classifier[1]. Frequency of the top 12 words is shown in the Figure 1 and it approximately follows a Zipfian structure as expected.

Along with ten basic genres, the dataset contains entries labelled as *Other* and *Not Available*. The decision was to throw out these instances since they do not contribute to the previously depicted objective. The last filtering step includes heuristic filtering based on compressibility of the lyrics. For each song we calculated the compression ratio using the Lempel-Ziv-Welch (LZW) compression algorithm (Welch, 1984).

We manually analyzed the content of the songs with the highest and the lowest compression ratio and decided to throw out all songs with compression ratio less than 0.02. After filtering steps, the dataset contains around 214.000 songs classified into 10 distinct genres. Preprocessing pipeline includes lowercasing and stopwords removal. The lyrical content is tokenized to form words and bigrams before the TF-IDF vectorization step.

Additionally, we conducted experiments on a smaller dataset with close to 50.000 instances. They were extracted from Million Song Dataset[2] as a subset called MusixMatch. This dataset contains lyrics and accompanying audio features which we used for a baseline model.

4. FastText Model

The problem we faced was how to properly vectorize a song so that it can be used by various machine learning algorithms. We wanted to generate a dataset of pairs, containing vector as an instance and genre as a label, to achieve a setup that can be used for training by any supervised algorithm. Vectors would be constructed using the lyrics of a song. As explained in section 3., the lyrical preprocessing had been done before engaging into the model itself. The initial idea was to simply use a word embedding algorithm on the entire song lyrics and then utilize a classifier such as SVM. The word embedding algorithm we chose to use was *fastText* (Bojanowski et al., 2016). It produces a 300-dimensional vector for an input word or n-gram. The results of using the entire song as a vector were abysmal due to its time consuming nature. Therefore, entire lyrics were not translated into the vector space, only the most important parts of it. The measure of "importance" that we used was the TF-IDF weighting scheme. Next step is to find the top 10 pairs consisting of weight and n-gram, and calculate the final song vector simply by reweighting the top 10 n-gram vectors by their TF-IDF scores and aggregating them.

This approach also leaves room for improvement since incorporating additional features is easy to do, merely by concatenating the 300-dimensional lyrics vector with any other numeric feature. Ones that have have a positive effect on the performances (ablation study may be considered) can be kept in the final model.

5. Repetition Model

Humans can easily recognize a repetitive song, but it is a difficult task for the computer because it lacks an appropriate measure. Using a compression algorithm to estimate repetitiveness of lyrics was proposed in (Morris, 2017). We tried this method on our dataset. Inspired by the sufficient differences in compressibility of lyrics by genres, we saw an opportunity to explore the possibility of informational gain in such phenomenon. Figure 2 shows a variation of this measure, size reduction of compressed lyrics in percentages, displayed on vertical axis. It is calculated as $1 - r_c$, where r_c is compression rate i.e., ratio of the size of compressed and uncompressed lyrics. *Hip-hop* genre noticeably departs from other ones. This can be supported by

[1] `https://code.google.com/archive/p/language-detection/`

[2] `http://millionsongdataset.com/`

the specificity of hip-hop lyrics as they are often the longest and can have intricate repetition structures.

5.1. Consensus Clustering

Traditional clustering algorithms suffer from several problems: random initialization of centers at the beginning can introduce a certain bias in results and moreover, validation of the number of clusters can pose an obstacle. Consensus clustering approaches mentioned problems as a method based on resampling. Predictions of a chosen clustering algorithm on a subsample (e.g., random 80% of data) are stored in multiple runs. In the end, for each entry (i, j), the calculated consensus matrix records the number of times items i and j are assigned to the same cluster divided by the total count of both items being selected (Monti et al., 2003). Let $D^{(1)}, D^{(2)}, \ldots, D^{(H)}$ be the list of H perturbed datasets, where H is the number of iterations. Entries of the connectivity matrix for each iteration $h \in \{1, 2, \ldots, H\}$ can be formulated as:

$$M^{(h)}(i,j) = \begin{cases} 1 & \text{if items } i \text{ and } j \text{ are in the same cluster,} \\ 0 & \text{otherwise.} \end{cases}$$

Finally, let $I^{(h)}$ be the indicator matrix such that the entry at position (i, j) is equal to 1 if both items are present in $D^{(h)}$, and 0 otherwise. The consensus matrix $\mathbf{C}$ can be denoted as follows:

$$\mathbf{C} = \frac{\sum_h M^{(h)}(i,j)}{\sum_h I^{(h)}(i,j)}.$$

In the final step, a standard clustering algorithm should be used once again to complete the clustering based on the calculated consensus matrix. Entries of the matrix $\mathbf{C}$ are now interpreted as similarity measures of a given pair.

5.2. Consensus Clustering as a Feature Selector

When crafting features for a machine learning task, one may be unsettled by the problem of estimating their importance i.e., quality. We chose a path of evaluating the results of clustering with features "on trial", hypothesizing that models which produce finely distributed clusters will also work well on the genre classification problem. This approach was used instead of standard feature selection methods to achieve a more robust estimation.

By introducing consensus clustering, we have a luxury of analyzing the resulting consensus matrix for each feature that is tested. The key to evaluate how well data was clustered lies in consensus distribution. Ideally, all consensus values should be either 1 or 0. This means that the algorithm was pretty decisive in every iteration and consistent throughout them. The best fit for number of clusters K can be found by studying the cumulative distribution function (CDF) of consensus which can be formulated as:

$$CDF(x) = \frac{\sum_{i<j} \mathbf{1}\{\mathbf{C}(i,j) \leq x\}}{N(N-1)},$$

where N denotes the number of rows (or columns since matrix $\mathbf{C}$ is symmetric). By inspecting the CDF shape and area under the curve, its bimodality can be assessed, which suggests the presence of clusters (Monti et al., 2003). The goal is to find the largest K that induces a large enough increase in the area under the CDF. We decided to pick K with the greatest area increase. In the final assessment of a feature, area under the CDF and its corresponding increase for the best fitted K are interpreted as their measurement of importance.

In the exhaustive process of crafting and selecting features that consider repetitiveness of song lyrics, we found that the LZW compression algorithm provides the best results. More specifically, the ratio of compressed lyrics size with LZW to original lyrics length was calculated. Other compression algorithms such as Huffman and LZ77 produced subpar scores when compared to LZW. Another idea was to create a matrix of co-occurrences. Entry at the position (i, j) is 1 if i-th word is the same as the j-th word, and 0 otherwise. Removal of punctuation and odd characters preceded the calculation. Lemmatization of given lyrics turned out to produce slightly better results, therefore it was kept in the final model. We tried to capture the structure of repetition within the co-occurrence matrix with minimal information loss. Standard matrix norms and custom made ones turned out to be poor choice for this task, so we tried a different approach described in the following subsection.

5.3. Approach by Convolutional Networks

Going one step further, we utilized a convolutional neural network (CNN) to recognize regularities in repetition structure of a specific genre. Co-occurrence matrix for each song described in the previous subsection is multiplied by the corresponding compression ratio and provided as an input to CNN. We padded the matrix with zeros to standardize its size to 1024×1024 (or trimmed it if it was larger), considering that the average word count of lyrics in used datasets was close to 1000. The architecture of the CNN was modeled on AlexNet (Krizhevsky et al., 2012). A simpler version was constructed with adapted number of parameters, primarily in the input layer. The training was enabled by adding a final softmax layer that was used to predict genre labels. Dimensionality of the output layer is changed correspondingly to the number of different genres in the given dataset.

The intuition behind this model is to connect certain patterns in repetition with rhythmic structure of a song. In other words, we presume that such correlation will be of utmost importance in genre classification task.

6. Results and Analysis

In the evaluation of results on MetroLyrics dataset, Natural Language Toolkit (NLTK) was used to provide a baseline model. Specifically, we chose the Multinomial Logistic Regression (MLR) as a classifier on top of a plain TF-IDF Vectorizer. Our models were evaluated using SVM classifiers. Test set contained 25% of the whole dataset which adds up to roughly 50.000 instances. Both the Repetition and the FastText model outperformed the baseline classifier. We supported this with t-tests (10-fold cross-validation) grounded on the micro F_1 scores. Test resulted in statistically significant difference with the significance level of 0.01. Performance scores are depicted in Table 1.

Table 1: Model performances on MetroLyrics Dataset

Model	micro-F_1
NLTK	0.192
Repetition Model	0.379
FastText Model	0.531

Regarding the comparison of our models, we showed that the FastText Model performs better than Repetition Model on MetroLyrics dataset with the same statistical setup. This is quite intuitive considering the larger informational gain of semantics in words. Nevertheless, the surprising fact is the ability of Repetition Model to cope even with plentiful genre types, which is 10 in this case.

We wanted to compare our Repetition Model with a system which is based on audio features. Implying hypothesis is that sound intuitively contains more information about genres than the lyrics. Custom model was built for that purpose. It bases its logic on features such as mel-frequency cepstral coefficients (Logan, 2000), spectral components (e.g., spectral rolloff, chroma, spectral contrast, etc.), tempo... *Librosa* library was utilized for extraction of mentioned features with songs' corresponding audio files provided as input. Calculated values are concatenated to form a numeric vector which is then forwarded to the classifier. Most of these features are widely used nowadays in various machine learning problems in the field of audio analysis (Darji, 2017).

Table 2 shows the comparison of the described baseline to Repetition Model by displaying their respective micro and macro F_1 measures. MLR was used as a final classifier to predict genres on test set (10.000 instances) of the MusixMatch dataset which contains four different genre types. The presumption that our Repetition Model can rank with the audio-based model in genre classification task is supported by the achieved results. It is hinted that the models produce roughly the same scores. We leave further experimenting with repetition structures as an opening to future work. Also, using larger corpora for training our models would very likely result in better performances. This behaviour was noticed when we experimented with training set sizes.

Repetition model performed extremely well in *hip-hop* songs. This hints how hip-hop songs are quite distinctive in their lyrical structure. We also noticed similarities between *pop* and *R&B* songs, which can be deducted from the confusion matrix displayed in Figure 3. Finally, constructed models provided decent results on used datasets and turned out to be a satisfying choice in classifying genres for lyrically dominant songs. It is due to notice the flaw of this approach, considering the complete inability to distinguish genre of purely instrumental songs. Despite this fact, it is not a worrying factor because of the abundance of lyrical songs which results in them largely outnumbering the instrumental ones.

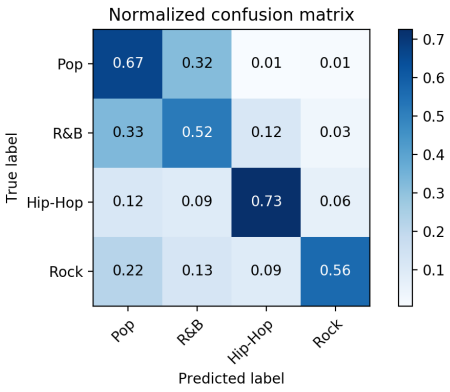

Figure 3: Normalized confusion matrix on MusixMatch dataset of the Repetition Model

Table 2: Audio Baseline and Repetition Model results on MusixMatch Dataset

Model	micro-F_1	macro-F_1
Audio Baseline	0.658	0.601
Repetition Model	0.662	0.609

7. Conclusion

Genre classification is an increasingly approached problem, mostly due to the fact of data explosion i.e., growth of online music databases. Though it is often tackled with audio analysis, some benefits like faster processing and easier access arise in studying the songs' lyrics. We presented two different models, achieving the best results with the FastText Model, but also gathering interesting insights of lyrical repetition by analyzing the Repetition Model.

The more usual approach depicted within the FastText Model turned out to have best performances. This is not a surprising fact since larger informational gain is available in the semantics of song lyrics than in the repetition structure itself. Nevertheless, grasping the phenomenon of repetitiveness in lyrics could prove to be very useful because of the growth of repetition both in quantity and importance. It can especially prosper in the analysis of genres that have strong bonds with lyrical structure.

References

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2016. Enriching word vectors with subword information. *arXiv preprint arXiv:1607.04606*.

Mittal Darji. 2017. Audio signal processing: A review of audio signal classification features. *International Journal of Scientific Research in Computer Science, Engineering and Information Technology*, 2:227–230, 05.

Robert O. Gjerdingen and David Perrott. 2008. Scanning the dial: The rapid recognition of music genres. *Journal of New Music Research*.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. 2012. Imagenet classification with deep convolu-

tional neural networks. In *Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'12, pages 1097–1105, USA. Curran Associates Inc.

Beth Logan. 2000. Mel frequency cepstral coefficients for music modeling. In *In International Symposium on Music Information Retrieval*.

Stefano Monti, Pablo Tamayo, Jill Mesirov, and Todd Golub. 2003. Consensus clustering: A resampling-based method for class discovery and visualization of gene expression microarray data. *Machine Learning*, 52(1):91–118, Jul.

Colin Morris. 2017. Are pop lyrics getting more repetitive? *The Pudding*.

Alexandros Tsaptsinos. 2017. Lyrics-based genre classification using a hierarchical attention network. *18th 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.

Terry A. Welch. 1984. A technique for high-performance data compression. *IEEE Computer*, pages 8–19.