Rhyming and Rhythmic Features for Musical Genre Classification by Lyric

Namdar Kabolinejad
namdar.kabolinejad@mail.mcgill.ca

Navin Kumar
navin.kumar@mail.mcgill.ca

Sung Jun Lee
sj.lee@mail.mcgill.ca

Abstract

In this report we describe the development of a suite of rhyme and rhythm-based features built for the purposes of musical genre classification via song lyrics. We use four standard language models to test the use of the features both on their own and as an augmentation to a binarized bag-of-words feature vector. We also describe the development and pre-processing of a custom dataset acquired by scraping LyricGenius for the top songs in the genres of rock, rap, pop and RnB.

1 Introduction

Text classification is a well studied area in NLP and its applications are both plentiful and varied in scope. MIR (Music Information Retrieval) researchers have long been interested in the task of musical genre classification. Classic methods for this task mainly consider acoustically-based features that can be extracted from audio. These include tempo estimation, average energy and spectral characteristics.

It has previously been shown that the combination of text and audio-based features can lead to greater success in this area compared to audio-based features alone. This is especially valuable in cases where the audio-based features are of low quality, such as when recording live audio from a mobile phone. In such cases, low-frequency energy is poorly captured, but lyrics may be discernable. Therefore, the development of lyric-based classifiers is an area worth pursuing, even if on its own it is unlikely to be more successful than classic acoustically-based MIR methods.

Because the text we are classifying is musical, we may exploit its unique qualities by considering its rhyming and rhythmic properties in addition to its lexical contents. While rhyme-based features have previously been studied to a degree, we have found no such research on the use of rhythmic features in this context. We hypothesize that lyrics from different genres should also display different rhythmic schemes within their lyrics, and that the addition of rhythmic features should thus improve performance in lyric-based genre classification. We also hope to improve results further by offering an expanded set of rhyme-based features compared to previous studies.

2 Related Work

Experts in MIR have suggested that current genre classification techniques can be bettered by using a variety of music representations rather than exclusively audio data (McKay and Fujinaga, 2006). Among the various alternative representations, lyrics are particularly appealing because they are easily attainable for most popular songs. As such there are a few studies that combine acoustic and lyrical features for genre classification (Mayer et al., 2008a).

While studies such as (Mayer et al., 2008a) document the success of combining lyric and audio-based features for genre classification, there are very few papers which document the development of the lyric-based features themselves. The only example we could find was (Mayer et al., 2008b), which describes the development of the features used in the authors' later study. Being one of the only works focused solely on lyric-based features, we base our project most heavily off of this paper. However, our project differs in some crucial aspects.

Mayer et al. (2008b) describe a suite of rhyme-based features that tests for four rhyme schemes: AA, AABB, ABAB, and ABBA. As described in 3.3 we expand on this with an additional four rhyme-based features. Additionally, for certain schemes we test for frequency rather than only presence. Moreover, our project introduces an additional dimension for lyric-based features in the form of rhythmic features. With the exception of the words-per-minute feature, rhythm-based features were not explored in the 2008 paper.

The use of rhythmic and cadential properties of song lyrics as features has been largely unexplored in the context of musical genre classification. However, these features have previously been analyzed in the context of poetry classification. This is unsurprising because many styles of poetry are rigidly defined by their rhythmic structure[1]. While genres of music are not generally *defined* by their cadential patterns in the way that poems are, we believe that rhythm still remains an important aspect to consider when investigating the internal structure of song lyrics.

3 Methodology

3.1 Building the Dataset

When searching for datasets suitable for this project it became clear that none of the sets readily available to us would suit our needs perfectly, as most did not include genre info at all. To address this, we elected to build our own dataset by scraping song lyrics from LyricsGenius.

We used a python wrapper for the LyricsGenius API[2] to scrape lyrics for the top 25 pages in the charts for pop, rock, R&B and rap. We also scraped lyrics for country music, but found that we could not obtain enough data samples for the category in comparison to the others.

3.2 Cleaning and Pre-Processing

After being scraped, the data is then cleaned by removing extra texts contained in brackets or curly braces, as these usually indicate the verse number of the song. We also remove parentheses but do not remove their contents, as these typically indicate the lyrics of backup singers.

After cleaning the data a *.txt* file is created for each genre containing the combination of all cleaned stanzas for the genre separated by whitespace. From this we filtered out samples which were less than 4 lines long to ensure that all musical features would be applicable. We also used langdetect to filter out any stanzas that were non-English. The lyrics were then parsed into CSVs such that each data sample would contain a single stanza of some song. The result of this filtering was a dataset of four categories, each with a minimum of 2800 text samples.

Before building the feature vectors, we use *prep.py* to load the data and create the training, development, and testing sets. While the file includes functions for lemmatization, stemming and stop-word removal, we elected not to use these as they would disrupt the musical features of the texts.

3.3 Design of Musical Features

In order to study these musical features we must first gather information on the phonetic and cadential characteristics of the words forming our text samples. To do this we replace every word token in the stanza with a tuple containing the word itself alongside a phonetic transcription, a count of the word's syllables and the stress pattern of the word formatted as a binary string. These are obtained using a python interface for the CMU pronouncing dictionary.

While the CMU pronouncing dictionary is vast, it is still relatively easy to find words that do not return any information [3]. Therefore we must consider how best to handle these out-of-vocabulary tokens.

To address this issue, we first attempt to recover the formal spelling of the word using a spell-checking tool. We use SymSpellPy to recover the most likely candidates for the word's formal spelling. We iterate through the candidates returned by SymSpellPy until one is found which has an entry in the CMU dictionary. We then replace the tuple for the out-of-vocabulary word with a tuple corresponding to this formal spelling.

This approach has the drawback of potentially altering the pronunciation of the word, and thus its phonetic transcription. Still, we find that in many cases the changes applied to the terms will not alter rhyming relationships[4]. It is still possible that SymSpellPy may fail to return any candidates which can be found in the CMU dictionary. In these cases we assign 'UNK' tags to the phonetic transcription,

[1] For instance, consider the rhythmic differences between limericks and sonnets.

[2] Our final version modified this library slightly by integrating the fix detailed here. To use the scraper it is necessary to use this fix and to supply a token for the Genius API.

[3] Typically this is because the word is either a slang term or because the word is written as an informal colloquialism. For instance, it is not uncommon to find the word "having" transcribed as " havin' ".

[4] For instance, if a stanza rhymes the words " runnin' " and " gunnin' ", the rhyme relationship will still be preserved when the words are replaced with "running" and "gunning" respectively.

number of syllables and stress-pattern of the tuple.

By collecting every line of tuples into a list, we obtain the data structure needed to analyze the musical features of a text sample. This process as well as the features detailed below are all implemented in *musical_features.py*.

3.3.1 Rhyming-Based Features

The pronouncing library implements a useful function that takes a word as input and returns a list of words that rhyme with it. Two words are considered to rhyme if their last syllables are composed of the same phonemes. With the exception of heteronyms we can also be certain that two words rhyme if they are represented by the exact same string. Thus in cases where a rhyme scheme simply reuses the same word we can even detect the rhyming of out-of-vocabulary words without a phonetic transcription.

While many complex rhyme schemes exist, we follow the work of (Mayer et al., 2008b) and concern ourselves only with rhymes involving the last word of each line in a stanza, all of which we collect into a new list. To detect rhyme schemes we then consider every sub-list of two, three, and four tuples. Using a mixture of frequency and binary formats we represent a total of 8 different rhyme schemes: AA, ABA, ABCA, ABAB, ABBA, AABB, ABAA, and AABA.

For the shorter and less strict rhyme schemes (AA, ABA, ABCA) we believe that the added granularity from measuring frequency may be worthwhile for better representing the overall rhyme scheme of the stanza. On the other hand, because the other rhyming schemes are longer and more specific, we imagine that a measurement of their frequency would be uninformative.

We complete our set of rhyme-based features by also measuring the frequency of rhymes that occur when all words are matched with one another pairwise.

3.3.2 Rhythm-Based Features

To represent the cadence of song lyrics, we simply concatenate the stress patterns of the words for every line in a given stanza. This gives us a rough approximation of the overall stress pattern for each line. These patterns can then be compared by taking their Jaro-Winkler similarity measure. We found Jaro-Winkler to be a useful similarity measure because it returns a value between 0 and 1 by design, making it easy to integrate into a feature vector.

We begin by measuring similarities of stress patterns within the stanza to one another. In a manner similar to rhyme schemes we take the similarity measures for every pair of adjacent couplets, couplets one line apart and couplets two lines apart. In each of these categories we take both the average similarity and the maximum for the stanza.

We also measure the similarity of each line against a set of 5 standard stress patterns that are used in western poetry. These include: Iambic Trimeter, Iambic Tetrameter, Iambic Pentameter, Trochaic Terameter and Trochaic Pentameter. Although these standard stress patterns are derived from classic western poetry, they are still reasonably common in western music and thus useful for our study. For each of these 5 patterns we again take the average and maximum of their similarities to every line of the stanza.

We complete our set of rhythm-based features by including a measure of “word complexity”. This is simply the total number of words divided by the total number of syllables in the stanza.

3.4 Training and Testing

For training, a variety of models and hyperparameters were tested. Using the preprocessed data splits (80:20) from *prep.py*, four baseline models were trained using a binary Bag of Words approach with Sklearn’s CountVectorizer: LogisticRegression (with CV eventually), LinearSVC, KNeighborsClassifier, BernoulliNB.

These four models were chosen as they easily support multiclass classification tasks and can support sparse matrices for quick computations. The training and test sets were prepared in similar proportions for each genre - around 2500 samples of each genre for the training set and around 600 samples of each genre for the test set. For each of the four classifiers, seven experimental feature vectors used to test our hypothesis.

5-Fold Cross Validation and hyperparameter tuning was done to obtain the highest performing classifier using the training set. For LogisticRegressionCV, various combinations of solvers (‘newton-cg’, ‘saga’, ‘lbfgs’) and max iterations (1000, 5000, 10000) were tested to derive our final models. For LinearSVC, various max iterations (1000, 5000, 10000) were also tested. For the KNN model, various numbers of neighbors (3, 5, 10) were tested. Various settings on the vectorizer were also tested to generate the baseline BoW features: TfidfVectorizer vs CountVectorizer, 1-gram vs 2-gram vs 1 and 2-gram mixed modeling. However, any method involving bigrams was computationally infeasible and also often led to convergence errors unless the max iterations were increased by multiple orders.

The scores for each combination of model and feature vector on the test set are shown in Table 1. Once the overall best performing combination (LogisticRegressionCV with BoW+Rhythm) was determined we generated its confusion matrix, precision, recall, and F1 scores for test set.

4 Results

The outcomes of our experiments yielded both optimistic and unexpected results. The first observable pattern is that the baseline model is quite successful; the best performing baseline model is more than twice as accurate compared to a random model. In fact, without the Bag of Words model, Table 1 shows that the accuracies drop significantly.

Additionally, it appears that the experimentative features were modestly successful in increasing model accuracy. Table 1 shows that the highest accuracies occurred for classifiers whose training data included the rhythmic and/or rhyme features. For three of the four classifiers, the BoW+rhythm features resulted in the highest accuracy. It appears that the addition of rhythmic features are the most effective, and that the rhyme features are actually either ineffective or negatively effective, with some models performing worse than the baseline when using the BoW+rhyme features.

This trend is accentuated when experimenting without the bag of words. Table 1 shows that working with only the rhyme features results accuracies slightly higher than random. However, when working with the rhythm features or a combination of both, the model's accuracies show a more noticeable 5-10% jump compared to a random model.

One interesting result is that using just the rhyme features results in a small increase compared to the random model whereas combining the bag of words with rhyme features sometimes leads to a decrease in accuracy compared to the baseline.

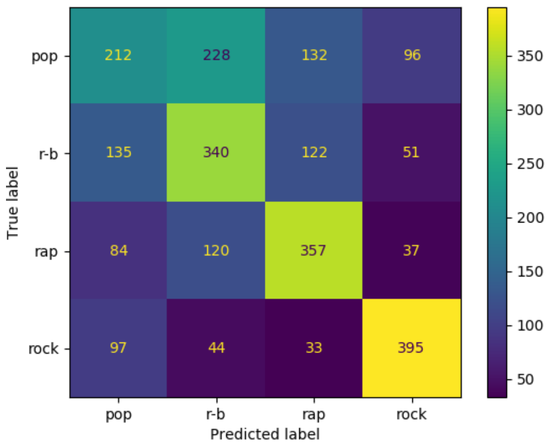
From the confusion matrix for the highest-accuracy model we see that pop is not easily distinguished from other genres. In fact, a pop song is more likely to be predicted as "r-b" than "pop" as shown in Figure 1. Pop also has the lowest precision, recall, and F1 scores out of any genre as shown in Table 2. On the other hand, the model excels at distinguishing rock from other genres. Most rock songs were labeled correctly and rock has the highest precision, recall, and F1 scores. There is thus a large difference of 0.37 in the F1 scores between rock and pop.

	Log. Reg.	**SVM**	**KNN**	**BNB**
BoW (Baseline)	0.519	0.499	0.404	0.498
BoW + Rhythm	**0.523**	**0.500**	0.414	**0.499**
BoW + Rhyme	0.516	0.496	0.414	0.497
BoW + Both	0.520	0.494	**0.421**	0.498
Rhythm	**0.300**	**0.336**	0.348	0.269
Rhyme	0.277	0.291	0.255	**0.277**
Rhythm + Rhyme	0.275	0.291	**0.356**	**0.277**

Table 1: Mean accuracy for each classifier. For each model the global maxima and maxima among only the musical feature vectors are in bold.

	Precision	**Recall**	**F1 Score**
Pop	0.38	0.30	0.33
RnB	0.45	0.53	0.49
Rap	0.57	0.60	0.58
Rock	0.70	0.71	0.70

Table 2: Precision, recall and F1 scores for Logistic Regression with BoW+Rhythm Model.



	pop	r-b	rap	rock
pop	212	228	132	96
r-b	135	340	122	51
rap	84	120	357	37
rock	97	44	33	395

Figure 1: Confusion matrix for Logistic Regression with BoW+Rhythm model.

5 Discussion & Conclusion

From our results we can conclude that our hypothesis is partially correct; augmenting a standard bag of words model with additional rhythmic features improves classification accuracy. However, our hypothesis was partially disproven as features based on rhyming schemes had little or even negative effect when appended to the baseline model. Unsurprisingly we find that rhythmic or rhyme-based features should be used only as an augmentation to boost accuracy rather than a stand-alone vector of features. Although our model is limited by the bag of words baseline, it would be interesting to test the augmentations on a bigram or trigram baseline which also captures some dependency information.

The success of rhythm-based augmentation demonstrates that different genres of music have quantifiably distinct rhythmic patterns, albeit at a modest level. Intuitively, this is a reasonable conclusion to make as when we examine the lyrical structure of songs, we expect to see a much higher syllable count or a unique cadence/meter for rap songs as opposed to RnB tracks for instance. As for the lack of success with the rhyme based features, it appears that these features may have been too ambiguous, which also may be an issue that is inherent to the songs themselves. Common rhyming schemes like AABB or ABAB are present throughout most contemporary western music and thus may not help for distinguishing genre. We are also unable to capture rhyming schemes where imperfect rhymes are used, as the CMU dictionary solely relies on precise phonetic transcription.

Another interesting conclusion that we see is that the novel set of features impact the model performance less when used as an augmentation versus by themselves. This may be due to the relative sizes of the feature vectors. The bag of words baseline creates a large feature vector when compared against the 27 musical features, meaning the influence of the musical features is small. This may explain the modest impact that the augmented vectors have. Thus when bigrams or even trigrams are used, it is expected that the presence of bigrams and trigrams will bring up the model accuracy but the augmentations will have an even smaller impact.

The last topic we want to address is the ambiguity of certain genres and particularly pop music. From Figure 1, we see that the model has a very difficult time classifying pop music. This problem is also present in RnB and rap, albeit at a smaller scale. This ambiguity however, may be an issue that is inherent to pop music itself, as it is largely an amalgamation of currently popular genres. With the recent rise in popularity of hip-hop, rap, and RnB, we expect that pop music would borrow many of their musical qualities. Whereas a song by Led Zeppelin can be easily classified as rock, a song by Bruno Mars could be classified under both RnB and pop, which may explain the numerous mislabels that occur for pop music.

This is also the case between rap and RnB. Songs by top rap artists such as Kanye West, Drake and Kendrick Lamar could easily also be classified under RnB as well. Thus the problem we aim to tackle may be one that is too ambiguous to begin with. As far as lyrical content, topics of money, fame, success, love, drugs are most commonly found in rap but certainly not unique to it. Additionally, while rhyme, rhythm and meter for each genre may differ slightly, there remains a large overlap between genres.

An additional investigation which may be worth exploring is multilabel multiclass classification instead of a single label multiclass classification, which would allow a song to be labeled under multiple genres instead of one. This would be more reflective of how many people think of musical genres in general, as it is common for people to regard a song as being multiple genres.

While our investigation did not greatly increase the accuracy compared to conventional methods, we were able to draw optimistic results and conclusions for feature vector augmentation based on rhythmic analyses of lyrics using NLP methods. Across multiple iterations of testing, we consistently found that augmented vectors resulted in slightly higher accuracies. We thus conclude that our novel augmentations were effective. We strongly believe that further exploration of our topic with the mentioned improvements will yield more dramatic results, and that genre classification for music can indeed be improved with NLP-based rhythmic analyses.

References

R. Mayer, R. Neumayer, and A. Rauber. 2008a. Combination of audio and lyrics features for genre classification in digital audio collections. *International Society for Music Information Retreival*, 12.

R. Mayer, R. Neumayer, and A. Rauber. 2008b. Rhyme and style features for musical genre classification by song lyrics. *International Society for Music Information Retreival*, 9.

C. McKay and I. Fujinaga. 2006. Musical genre classification: Is it worth pursuing and how can it be improved?