Genre Classification using Song Lyrics

Kruthika Vishwanath
Erik Jonsson School of Engineering and Computer Science
University of Texas at Dallas
kxv150930@utdallas.edu

***Abstract*—Genre classification techniques combining lyric and audio features for Music Information Retrieval (MIR) have been studied widely in recent years. This work investigates the performances of musical classification using lyric features only. In this study, the Part-of-Speech (POS) and Word2vec features were utilized for genre classification of a collection of 1046 songs. Five musical genre categories named 'Hip-Hop', 'Rock', 'Pop', 'Christian', 'Hip Hop/Rap', 'Dance', 'Latin' were selected.Experiments show that POS Tagging feature outperforms Word2vec when Gradient Boosting Classifier is being used to classify genre based on song lyrics.**

***Keywords*—Classification; Music Information Retrieval; Musical Genre; Lyrics Analysis; POS-tagging; Word2vec**

I. Introduction

There has been ever increasing demand to listen to online music. Due to which, having recommendation and retrieval system becomes vital to give best experience for the customers (example: in case of users on Spotify). This definitely helps the music to be reached out to larger audience, benefitting the music industry along with big streaming services like Soundcloud and Spotify. In principle, audio and textual can be used to analyse the music piece. But, in this project, focus is on lyrics of the song using POS tagging and Word2vec. Classification of song is majorly researched topic in NLP. Genre Classification itself is a Natural Language Processing problem. The end goal of NLP is to extract meaning from the text. In genre classification, this relates to finding features to classify music using lyrics. Song lyrics exhibit a certain structure, as they are organized in blocks of choruses and verses. Many songs are organized in rhymes, patterns which are reflected in a songs lyrics and these structures in text are much easier to detect rather than audio.[13] This paper is organized as follows: Section 2 reviews related work on lyric analysis, musical genre classification in Natural Language Progressing.Section 3 introduces the dataset creation and data preprocessing for this project.Section 4 describes lyrics features examined.Section 5 describes the experiments. Section 6 presents the conclusions and future work.

II. Related Work

Several research teams have worked on adding textual information to the retrieval process, predominantly in the form of song lyrics. One of them [13] used Part-of-Speech (POS) feature for classification of a collection of 600 songs. Ten musical genre and mood categories were selected respectively based on a summary from the literature. They came up with experiments which showed that classification accuracies for mood categories outperform genres. Whereas, Michael Fell and Caroline Sporleder [10] convey by providing a novel approach for analysing and classifying lyrics, experimenting both with n-gram models and more sophisticated features that model different dimensions of a song text, such as vocabulary, style, semantics, orientation towards the world, and song structure.There has not only been work going on using textual lyrics, but, audio as well. Building ensembles of audio and lyrics features to improve musical genre classification by [12] experimented with a small dataset of 3000 songs and used ensemble methods to choose the best performing feature set and classification algorithm. From [11] one can learn the usage of various machine learning algorithms, including k-nearest neighbor (kNN),k-means, multi-class SVM, and neural networks to classify the following four genres: classical,jazz, metal, and pop.

III. Data Pre-Processing

A total of 1046, names of songs along with lyrics, genre, artist name, album name were collected by scrapping www.songlyrics.com and www.metrolyrics.com using BeautifulSoup. BeautifulSoup is Python package for parsing HTML and XML documents [2].Those genres were collected, which had large collection of songs. Genres named 'Hip-Hop', 'Rock', 'Pop', 'Christian', 'Hip Hop/Rap', 'Dance', 'Latin' were a few of them. On analyzing the data collected, non-English songs were found, for which, I decided to remove them as it had less data in train data at the time of classification. Further cleaning such as removing, stop words using nltk, brackets using regular expressions and words like 'Verse', 'Chorus' from the lyrics, aided in gathering clean data, which will be used for classification and feature collection.[7]

IV. Feature Selection

Lyrics are a very rich resource and many types of textual features can be extracted from them. Since they are usually short, every word becomes important in the topic such as genre classification[13]. From [5] & [10] one can learn that feature selection can be made from Bag of Words, Word Endings, Line Length, Number of Lines in a song, Punctuation, Part-of-Speech, Repetitive Structures in the song and Chorus of the song. Also, Past studies on authorship attribution have shown that common function words such as 'of' can be an effective marker of author style. Differentiating between styles of authorship in songs may also help us differentiate between genres, since writers in a single genre will probably have more similar styles than between genres. On the other hand, content words, allow us to extract semantic meaning from the song. Words such as 'life' or 'love' can be strong indicators of the song's topic. By using bag of words, we can compare tendencies between different genres. While the endings of words can indicate things such as verb tense, meaningful

suffixes, and slang use. All of these things combined contribute to the semantic meaning of a song as well as the writing style of a song. Surprisingly, the length of a line in the song can indicate several things dealing with the rhythm and pattern of a songs acoustics.Similar to line length, the number of lines can also hint at the acoustics behind the song lyrics. One can view the number of lines as the inverse of the line length; more lines may indicate a style with frequent stops, while fewer may indicate longer lines or drawn out words. For example, frequent line breaks can indicate a song which flows in a much more choppy way than long lines with many words. The use of punctuation marks can reveal both lexical and acoustic information. The use of things apostrophes can indicate the use of slang and shorthand, and can be seen as a marker for an artists style of speech. The use of periods, commas and other sentence delimiters however, indicate stops in the song. These can reveal the songs pattern of rhythm. [5] has shown that part-of-speech statistics are also useful feature in analysis of authorship. Often Repetition plays a large role in the musical character of a song.To score repetition, we simply count the number of words that are repeated. Thus, a single word repeated 5 times will add a count of 4, while a 4-word phrase repeated once will also add a count of 4. We then normalize the score against the total number of words.In addition to repetition in the entire song, we can also point out repetitions within a line. Generally, these have a different meaning than repetitions across a song; instead of being a repeated phrase or verse, these can be single words sung in multiples. For example, Matchbox 20 sings Baby, baby, baby when all our love is gone These types of repeats are an indicator of artist style, and thus could also distinguish between genre styles. Given specified time and keeping concepts taught in class in mind, I focused on Part-of-Speech(POS) tagging frequencies in the lyrics of a song and it's word2vec vector values as features in this work. [7].

A. *POS tags*

Lyrics are a very rich resource as many features such as sentiments, genre of the song, artist name, album name can be extracted. In this work, from the most commonly used feature types in related text classification tasks, I focused on lyric feature which was Part-of-Speech. Part-of-speech (POS) tagging, also called word-category disambiguation is a lexical categorization or grammatical tagging of words according to their definition and the textual context they seem [4]. Basic POS categories are for example nouns, verbs, conjunction, articles, and adjectives. In [13], Rauber and Mayer presume that different genres will differ in the category of words used. In their study, 9 POS categories which include nouns, verbs, pronouns, relational pronouns, prepositions, adverbs, articles, modals and adjectives were used. I similarly used the same POS categories. and another POS category, interjections. Interjections were chosen because it can express emotion [13]. In this work, I fed song lyrics to NLTK's POS tagger to get the parts of speech tags of lyrics of each song. The Natural Language Toolkit, or more commonly NLTK, is a suite of libraries and programs for symbolic and statistical natural language processing for English written in the Python programming language [3]. All the 1046 lyric text with POS tagger were then analyzed in terms of the occurrence of each unique word in each different category of POS. Further, I normalized the occurrences and used it as a feature. The following graphs shows the number of different pos tags per genre.

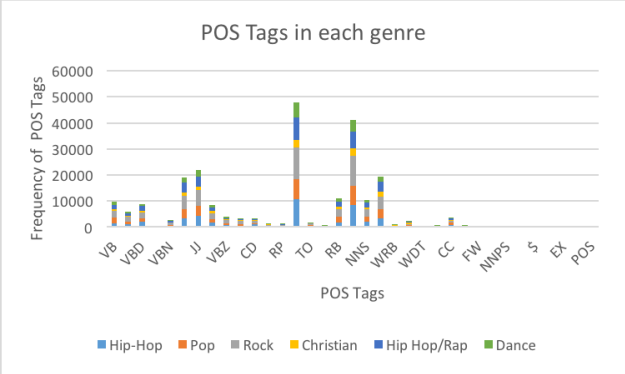

Fig. 1. Number of different POS tags per genre

B. *Word2vec*

Word2vec plays very important role in genre classification due to its ability to represent both semantic & syntactic meaning of a word by representing them as vectors of numbers obtained during the training phase of the models [8]. The idea of word2vec is to represent words using the surrounding words of that word. For example, assume that we have the following sentence:"*I like playing X*". Here, you may not know the meaning of X, but you for sure know that "X" is something one can play with, and also, it is something that can be enjoyable to some people. Hence, our brain reaches to this conclusion after reviewing the Xs surrounding words, "like" and "playing". The target of each model under Word2vec is to produce fixed-size vectors for the corpus words, so that the words which have similar or close meaning have close vectors. This is the idea of inventing word2vec technique [8]. In this work, I trained a word2vec model on entire collection of 1046 songs dataset along with brown corpus. Following which, I generated the word2vec vector of each unique token in each song and used Frequency based Embedding technique by taking average of all the vectors to get a word2vec vector of a song. Fig.2. shows the glimpse of how the word2vec of each song(indexed) looks. In this work, this is considered as Word2vec features which is further used for classification. Frequency based Embedding is word embedding which learns a vocabulary from all of the documents, then models each document by counting the number of times each word appears [9].

	A	B	C	D	E	F	G	H	I
1	w2v_1	w2v_2	w2v_3	w2v_4	w2v_5	w2v_6	w2v_7	w2v_8	w2v_9
2	-1.0878004	0.12304951	0.06552577	-0.5536359	-0.2606554	-0.7704756	-0.3779966	-0.3040542	0.56207883
3	-0.8375793	0.04552137	0.05470621	-0.4939846	-0.2646846	-0.5621272	-0.2496583	-0.2802084	0.38711825
4	-1.1009521	-0.0100354	0.0179393	-0.520453	-0.2763083	-0.7224665	-0.2881601	-0.3077981	0.46160442
5	-1.2809091	-0.1152255	0.07014985	-0.5463694	-0.3204405	-0.7390555	-0.2151215	-0.3616564	0.29239428
6	-0.8958178	-0.0021794	0.09497304	-0.506143	-0.1873789	-0.7169939	-0.1395657	-0.3859912	0.22919774
7	-0.9760765	-0.1088514	0.00085169	-0.4538344	-0.31738	-0.5283087	-0.2253862	-0.2836718	0.30119517
8	-0.5479242	-0.280407	-0.0579549	-0.485848	0.02205871	-0.3438337	-0.126119	-0.5137306	0.06316466
9	-1.2731377	-0.1165427	0.07085545	-0.5491124	-0.3177544	-0.7368356	-0.2085405	-0.3628921	0.28648162
10	-1.2055124	0.05878423	-0.0166581	-0.5310739	-0.3236113	-0.745338	-0.4738258	-0.2800212	0.70258707
11	-0.8451545	-0.0303995	0.00723589	-0.5210829	-0.2625451	-0.5086398	-0.1981357	-0.3915914	0.35435611
12	-0.8716611	-0.0786479	0.0141706	-0.4257225	-0.1824914	-0.4984823	-0.1817135	-0.3171009	0.25067666
13	-0.9053043	-0.053554	0.03482655	-0.4707633	-0.3006245	-0.5394156	-0.2147019	-0.3562807	0.30283853
14	-0.8903921	-0.0029486	0.08982042	-0.5033686	-0.2042811	-0.7274706	-0.203705	-0.3569284	0.28078005

Fig. 2. Word2vec vector of each song

V. Experiments

For the experiments, I employed the scikit-learn toolkit to utilize classifiers such as Logistic Regression, Support Vector Machine, Gradient Boosting, Random Forest Classifier and evaluated them by calculating metrics like Accuracy, F1 Score, Precision, Recall [1]. Scikit-learn is a free software machine learning library for the Python programming language [6]. It features various classification, regression and clustering algorithms including support vector machines, random forests, gradient boosting, k-means and DBSCAN, and is designed to interoperate with the Python numerical and scientific libraries NumPy and SciPy[11]. The classification experiments were first performed individually on POS features and word2vec features followed by, combining POS and word2vec features.

A. POS features

Using the frequency of each POS tag in lyrics of each song[Fig. 1.], I proceeded with normalizing the frequency using the following formula :

$$pos\_features\_normalized = \frac{(x - min(x))}{(max(x) - min(x)} \quad (1)$$

where, x = occurrences of each POS tag in each of the lyrics of the song.

Some of the trivial challenges faced with the data before normalization was missing data. I overcame it by filling the missing values with mean of the x (occurrences of each POS tag in each of the lyrics of the song). Further going ahead, I split pos_features_normalized values into 70% train & 30%test data before feeding it to the machine learning models such as Logistic Regression, Support Vector Machine, Gradient Boosting Classifier & Random Forest Classifier.

Model	Accuracy	F1 Score	Precision	Recall	Parameters
Logistic Regression	0.5468	0.5282	0.5623	0.5278	C=1e5
Support Vector Machine	0.3444	0.1518	0.3873	0.2097	Default
Gradient Boosting	**0.9939**	0.9937	0.9924	0.9955	max_depth=20
Random Forest	0.8006	0.8046	0.8076	0.8076	max_depth=20

TABLE I. Metrics value using POS features only

This experiment showed that Gradient Boosting Classifier, fed with POS Tags frequencies, outperformed other classifiers used in this work with an accuracy of 99.39%. Following table shows the value of the metrics used to evaluate the performance of the classifiers.

B. Word2vec features

Similar to previous experiment, wherein, only POS Tag frequency was used, in this experiment, I fed the Word2vec values obtained into each of the classifiers, Logistic Regression, Support Vector Machine, Gradient Boosting & Random Forest. Once again evaluated the performance of each model using Accuracy, F1 Score, Precision & Recall with 70% train data and 30% test data. Following is the table which gives the values of the metrics used to evaluate each of the classifier :

Model	Accuracy	F1 Score	Precision	Recall	Parameters
Logistic Regression	**0.5740**	0.5325	0.5641	0.5233	C=1e5
Support Vector Machine	0.3202	0.1016	0.1268	0.1792	Default
Gradient Boosting	0.3927	0.3564	0.3853	0.3458	max_depth=20
Random Forest	0.4320	0.3729	0.4399	0.3678	max_depth=20

TABLE II. Metrics value using word2vec features only

This experiment resulted in Logistic Regression, fed with, Word2vec values outperforming rest of the classifiers with an accuracy of 57.4%.

C. Combination of POS & Word2vec features

In this experiment, I concatenated, normalized POS Tags frequency values with Word2vec values and fed into each of the classifiers (Logistic Regression, Support Vector Machine, Gradient Boosting & Random Forest) used in this work. Once again used 70% of the data as train and rest 30% as test. Following table shows the evaluated values measuring the performance of each model.

Model	Accuracy	F1 Score	Precision	Recall	Parameters
Logistic Regression	0.6888	0.6968	0.7007	0.6955	C=1e5
Support Vector Machine	0.3504	0.1474	0.5546	0.2020	Default
Gradient Boosting	**0.9667**	0.9693	0.9742	0.9665	max_depth=20
Random Forest	0.7371	0.7177	0.7423	0.7054	max_depth=20

TABLE III. Metrics value using POS+word2vec features

This experiment resulted in Gradient Boosting, fed with, concatenation of POS Tag Frequencies and Word2vec, outperforming rest of the classifiers with an accuracy of 96.67%.

D. Comparing results

To understand among which of the features (POS Tag ONLY, Word2vec ONLY, Concatenation of POS & Word2vec) gave better accuracy, following graph Fig. 3. was plotted with Accuracy taking over Y-axis and three features discussed above, taking over X-axis. We can infer from the graph that, POS Tag ONLY and Concatenation of POS Tag and Word2vec outperformed Word2vec ONLY in terms of accuracy. With Gradient Boosting taking over Random Forest followed by Logictic Regression and Support Vector Machine.

VI. Conclusion and Future Work

In this work, I showed the feasibility of POS & Word2vec in genre classification using song lyrics. Comparison of the accuracy for genre classification using POS & Word2vec shows that POS performed better than Word2vec in genre classification. I believe the accuracy increased wherever POS Tagging was deployed. This may be due to the better precision offered by NLTK toolkit while tagging, over the biggest problem with word2vec's inability to handle unknown or out-of-vocabulary (OOV) words. If the model hasnt encountered

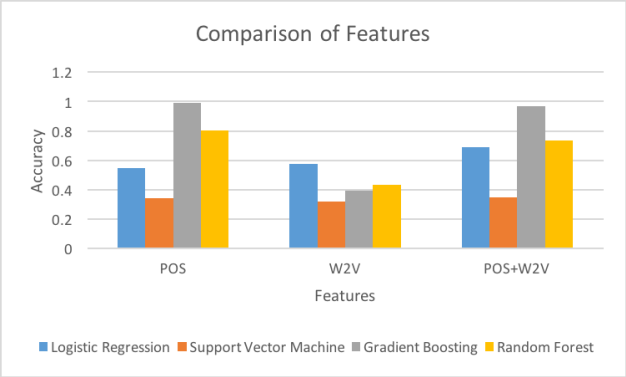

Fig. 3. Features v/s Accuracy using different classifiers

a word before, it will have no idea how to interpret it or how to build a vector for it. In this case, one is forced to use a random vector, which is far from ideal [9]. In future, I plan to improve this work in the following ways: (1) In order to achieve better result, other important lyric features such as length of the sentence, Rhyme features, Repetitive structures can be considered. [10]. (2) Work on a larger data collection

ACKNOWLEDGMENT

I would like to thank Professor Dan Moldovan, Natural Language Processing professor at University of Texas at Dallas, for providing me direction and motivation to improve results presented in this work. Also, I would like to thank my classmates from Natural Language Processing class at University of Texas at Dallas, Unnati Singh & Parag Dakle for helping me understand some aspects of pandas library, a software library written for the Python programming language for data manipulation and analysis.

REFERENCES

[1] Accuracy, precision, recall & f1 score: Interpretation of performance measures - exsilio blog. http://blog.exsilio.com/all/accuracy-precision-recall-f1-score-interpretation-of-performance-measures/. (Accessed on 12/01/2017).

[2] Beautiful soup (html parser) - wikipedia. https://en.wikipedia.org/wiki/Beautiful$_{S}oup_{(}HTML_{p}arser).(Accessedon12/01/2017)$.

[3] Natural language toolkit - wikipedia. https://en.wikipedia.org/wiki/Natural$_{L}anguage_{T}oolkit.(Accessedon12/01/2017)$.

[4] Part-of-speech tagging - wikipedia. https://en.wikipedia.org/wiki/Part-of-speech$_{t}agging.(Accessedon12/01/2017)$.

[5] sadovsky-x1n9-1-224n_final_report.pdf. https://nlp.stanford.edu/courses/cs224n/2006/fp/sadovsky-x1n9-1-224n$_{f}inal_{r}eport.pdf.(Accessedon12/01/2017)$.

[6] scikit-learn - wikipedia. https://en.wikipedia.org/wiki/Scikit-learn. (Accessed on 12/01/2017).

[7] Song-genre-classification/report.pdf at master ppatel104/song-genre-classification. https://github.com/ppatel104/Song-Genre-Classification/blob/master/report.pdf. (Accessed on 12/01/2017).

[8] [thesis tutorials i] understanding word2vec for word embedding i ah's blog. https://ahmedhanibrahim.wordpress.com/2017/04/25/thesis-tutorials-i-understanding-word2vec-for-word-embedding-i/. (Accessed on 12/01/2017).

[9] Word embedding data science group, iitr medium. https://medium.com/data-science-group-iitr/word-embedding-2d05d270b285. (Accessed on 12/01/2017).

[10] Michael Fell and Caroline Sporleder. Lyrics-based analysis and classification of music. In *COLING*, 2014.

[11] A. Karatana and O. Yildiz. Music genre classification with machine learning techniques. In *2017 25th Signal Processing and Communications Applications Conference (SIU)*, pages 1–4, May 2017.

[12] R. Mayer and A. Rauber. Building ensembles of audio and lyrics features to improve musical genre classification. In *2010 International Conference on Distributed Frameworks for Multimedia Applications*, pages 1–6, Aug 2010.

[13] Teh Chao Ying, S. Doraisamy, and Lili Nurliyana Abdullah. Genre and mood classification using lyric features. In *2012 International Conference on Information Retrieval Knowledge Management*, pages 260–263, March 2012.