

# Music Genre Classification using Lyric Mining Based on tf-Idf

Meenakshi K, SafaM., Geetha G., Saranya G., SundaraKanchanaJ

**Abstract.** With the advancement in the internet technologies, music domain has flourished with better access to various music libraries. In the present times, we are able to access music files over the internet with ease. Nowadays, the lyrics sets are categorised into different genres which cater to various listener moods. Users prefer to listen to music that best suits their mood. Thus considering the need for such classification, research works are being carried out to develop methodologies that can distinguish the music based on individual mood. In this research, instead of using the traditional method of audio feature analysis, we propose to develop a system which analyses the lyrics dataset of the songs based on the features extracted from the training phase and we can predict the mood of the song that is presented to the system at the validation stage. The proposed system is considering five moods containing one hundred songs each, for the validation purpose. The system is capable of predicting the mood of the song based on the analysis of the lyric text.

**Keywords:** classification, Lyric mining, Music Genre Classification, , tf-idf,

## I. INTRODUCTION

With the growth of technology, a large amount of music related data are generated and these data are stored in using efficient storage technologies which make possible for the faster retrieval of data. These data can be the metadata of the songs, the audio files, the video files and the lyrics. The music is generally categorized based on genres and moods for the people who have selective preference in music. Due to easy accessibility of music in the present time, people choose to listen to music that specifically caters to their mood of the moment such as happy, sad, angry, relaxed, calm, romantic, gloomy, energetic, confident etc. Thus advancement is being made to develop Music Information Retrieval (MIR) techniques to derive information from the lyrics and categories the music according to the mood.

Revised Manuscript Received on January 15, 2020

Corresponding author:

**Meenakshi K**, Department of Information Technology SRM Institute of Science and Technology Chennai, India

**SafaM.**, Department of Information Technology SRM Institute of Science and Technology Chennai, India

**Geetha G.**, Department of Information Technology SRM Institute of Science and Technology Chennai, India

**Saranya G**, Department of Information Technology SRM Institute of Science and Technology Chennai, India

**SundaraKanchanaJ**, Department of Information Technology SRM Institute of Science and Technology Chennai, India

In the proposed system, *tf-idf* and *tf\*tf-idf* weighing schemes are used to determine the importance of the words in the lyric text with the corresponding moods. Here *tf* and *idf* stand for term and inverse document frequencies respectively and are the metrics used to determine the usefulness of the words in determining the mood of the songs. Then to improve upon the existing algorithm, a new weighing scheme *tf\*tf-idf* is introduced whose sole purpose is to improve the existing weights of the words in the lyric text. Thus on comparing both algorithms, it is found that the existing *tf-idf* values have given better performance and also there is a slight improvement observed in the accuracy for the correct prediction of lyric text.

## II. RELATED WORKS

Menno van Zaanen, Pieter Kanters [1] in their study focuses on using features extracted from the lyrics to classify the lyrics in various moods and genres. They used various measures like K-Nearest Neighbour (KNN), Naïve Bayes (NB) and Support Vector Machine (SVM) to compare the accuracies of genres and other moods. In their study, they had found that happy and sad had better accuracies than calm and aggressive moods.

VipinKumar, and SonajhariaMinz [2] in their study used the Sentiword net to extract features related to sentiments of the words in the lyrics. They used three sentiment features i.e., ratio of the positive and negative scores, normalized ratio, and average of the positive and negative scores. Through their study, they were able to achieve an accuracy of 72% using Naïve Bayes classifier.

Teh Chao Ying, ShyamalaDoraisamy and LiliNurliyana Abdullah [3] have studied as how to determine the mood of the song by utilizing the lyrical features of the songs. In this study they have used another feature known as Part-Of-Speech (POS) for the classification of the collection of the songs. They calculated various classification parameter values for the genres and moods. From their study they found out that mood classification was better than genre classification for POS. The traditional method of determining the mood or genre of the song was through audio analysis. But with a lot of research works that were being carried out, many researchers tried to use both lyrics and audio to improve the accuracy of the mood prediction. One such study has been done by X. Hu, and J.S. Downie [4]. Xiao Hu J. Stephen Downie developed the work on multi modal mood classification. They combined both audio and video classification. [5]. X. Hu, J.S. Downie and A.F. Ehmann examined the purpose of lyric text. They developed model that classify a song from the large ground truth set of 5,585 songs and 18 mood categories based on social tags [6]. R. Mayer, R. Neumayer, and A. Rauber were proposed the song classification based on frequency of certain rhyme patterns, POS features and



static features such as Words per minute (wpm) [7].

The n-gram framework model was used for extracting the semantic features from lyric text. And the classification accuracy was compared by supervised learning approach (Naïve Bayes, Support vector machine and maximum entropy) [8]. Fuzzy clustering technique was used to detect the emotions of songs based on lyrics. The approach determine the emotion of Chinese lyrics by using emotion lexicon, called ANCW [9]. Teh Chao Ying, Shyamala Doraisamy and Lili Nurliyana Abdullah [10] in their study used the method of analysis of the lyrics and based on the analysis, they tried to observe interdependency between genre and mood. They studied many weighing schemes and compared the results to determine the most efficient weighing algorithm. The Tamil Lyrics were classified by using tf-idf score by Kanchana et.al.[11]. In their study, the lyrics are categorized in two genres. Vishnuprya et.al., classifies the music into various genres by extracting the feature vector and Mel Frequency Cepstral Coefficient (MFCC) is used as a feature vector for sound sample. [12]

### III. METHODOLOGY

The basic approach for the classification of songs based on lyrics involves the following steps or phases.

1. Data collection phase
2. Training phase
3. Validation phase

#### A. Data collection

1. **Test Dataset collection:** In the process of music information retrieval, the first step is raw data collection. There are a number of mood categories according to which the songs can be classified and they range from ten to twenty moods. For our system, we have chosen basically five mood categories namely happy, sad, romantic, angry and spiritual as other categories can be broadly classified under these five major categories. Under every mood, we have selected a dataset of 100 songs.

For the classification of the songs into different moods, we selected two online sources which offer song preferences based on mood. For each song in the dataset, mood labels are compared in both the sources and then classified them into their corresponding moods. Thus, a total of 500 songs are placed in text file for training the system and 50 songs are set aside for the validation phase.

2. **Data preprocessing:** The data preprocessing step basically involved the sampling of data i.e. making our test dataset ready for the training phase. Every song that we had downloaded, involved phrases such as “[Intro:]” “[Verse:]”, “[Chorus:]” and “[Bridge:]” which were insignificant and had to be removed manually. A Java program was used to remove all the punctuation marks that our text files consisted. The data cleaning process is done manually.

#### B. Training phase

1. **Data pre-processing for training:** Once the data has been removed of any irregularities as given in the data preprocessing step, the next step involves the pre processing

step for the training phase. This step basically involves three major processes:-

2. **Tokenization:** Tokenization process involves breaking the string of the lyrics into tokens. This can be carried out based on a space, a character or by a string. This step is necessary because in the reading process, the words are read word by word. The process is done through a program written in Java.

Example: “This is an example string” can be tokenized into ‘This’, ‘is’, ‘an’, ‘example’ and string”

3. **Stop word removal:** Function words such as ‘a’, ‘the’, ‘that’, ‘and’ etc., do not carry any meanings thus are of little help in determining the *tf-idf* values and eventually in the classification. Thus, they had to be removed. This removal is carried out by a stop word removal program written in Java. Example: “The heart is a bloom Shoots up through the stony ground There's no room No space to rent in this town”. The italicized words are stop words thus can be removed. Thus the result we have after removing the stop words is “heart bloom Shoots stony ground room space rent town”.

4. **Morphological analysis:** The morphological analysis process also called as stemming process reduces the words onto their root form. For the *tf-idf* value calculation, the nouns are more significant than the verbs. For the process of stemming, we had used the Lancaster algorithm. For example ‘swimming’ reduces to ‘swim’.

#### C. tf-idf weighing scheme algorithm

Bag of Words (BOG) method is used for *tf-idf* calculation. In this method, after stemming the words in the lyric, dataset are put into a single text document. This document acts as a bag of words helps further in the calculation of the *tf*. In our proposed system five BOG documents are available corresponding to the five moods. In the *tf-idf* weighing scheme, the relative importance of the words are studied with respect to a particular mood. The *tf-idf* is calculated for each word.

Let the BOG document be denoted by *d*. The term frequency (*tf*) of the term *t* gives the value which tells about how frequent the term occurs in the document *d*. *tf* is defined as the number of occurrences (*n*) of the term *t* in the document *d* divided by the total number of the occurrences of all the terms (*N*) in the document *d*.

$$tf_{(t,d)} = \frac{n}{N} \quad (1)$$

where

*n* = number of occurrences of the term *t* in the doc *d*

and

*N* = the total number of the occurrences of all the terms in the document *d*.

*idf* stands for inverse document frequency.

*idf* of the term *t* is defined as the logarithm of the value obtained by dividing the total number of documents (*N<sub>D</sub>*) by the number of documents (*N<sub>T</sub>*) in which the term *t* has appeared. i.e.,

$$idf_{(t,d)} = \log \left( \frac{N_D}{N_T} \right) (2)$$

where

*N<sub>D</sub>* = the total number of documents and

$N_t$  = the number of documents in which the term  $t$  has appeared. is calculated by extracting the  $tf-idf$  values from the corresponding  $tf-idf$  document of the mood.

Once we have obtained the  $tf$  and  $idf$  values of the term  $t$ , then  $tf-idf$  weight can be obtained by multiplying both the values  $tf$  and  $idf$  using Equation (3)

$$w(tf-idf) = tf * tf-idf(3)$$

Thus we have  $tf-idf$  weights for each term in the BOG document corresponding to each mood. If we have higher value of a term for a mood, which implies that the word holds higher importance to that particular mood. For some words, the value is zero too. That basically implies that the word is common among all the documents and thus the  $idf$  value becomes zero. The  $tf-idf$  values for the terms corresponding to their mood are copied and pasted on to separate text files. These text files are manipulated later in the validation phase using Java programs to determine the  $tf-idf$  values of the terms of the test lyric file and thus determining the mood of the lyrics.

#### Dtf \*tf-idf weighing scheme algorithm

In the  $tf * tf-idf$  weighing scheme algorithm, we attempt to improve our values corresponding to the terms that hold importance to the moods. In this method, we first calculate the term frequencies of the terms of the test lyrics. Then for each term we extract the  $tf-idf$  values of the terms from the  $tf-idf$  documents corresponding to each mood. We compare the values and determine the greatest amongst all of them. Then corresponding to the mood to which the greatest value belongs, we multiply the  $tf$  by the  $tf-idf$  and update the value. These updated values are used in the calculation of the result. The values obtained after updating are added and summed up. Average value with respect to the five moods is calculated and the mood with highest value is declared to be the mood of the test lyrics. For validation, let  $t1$  be a term that exists in the file and its term frequency  $tf$  in the file be 10. Corresponding to the term  $t1$  the respective  $tf-idf$  values are extracted from the  $tf-idf$  documents. Suppose the values for the term  $t1$  are:

$$\{tf-idf_{(happy)} = 0.014934, \\ tf-idf_{(sad)} = 0.0179584, \\ tf-idf_{(romantic)} = 0.073479, \\ tf-idf_{(angry)} = 0.0156002 \text{ and} \\ tf-idf_{(spiritual)} = 0.01034\}$$

It is observed that  $tf-idf_{(romantic)}$  has the highest value. The highest value  $tf-idf_{(romantic)}$  is multiplied by the term frequency  $tf$  of  $t1$  (10). Thus,  $tf-idf_{(romantic)}$  value \*  $tf = 0.073479 * 10 = 0.73479$ . This value alone is updated in the existing  $tf-idf$  values in the document.  $\{tf-idf_{(happy)} = 0.014934, tf-idf_{(sad)} = 0.0179584, tf-idf_{(romantic)} = 0.73479, tf-idf_{(angry)} = 0.0156002$  and  $tf-idf_{(spiritual)} = 0.01034\}$ . The same method is applied for the rest of the terms in the test lyric file.

Then the normal validation process is implemented to determine the mood of the song.

#### E. Validation phase

For the validation of a test lyric file, a Java program is used which would read the words one by one and would extract the corresponding  $tf-idf$  value of the word for every mood. These values are added and the total value of the  $tf-idf$  of the terms in the test document is calculated. For example,  $t_1, t_2, t_3, \dots, t_n$  are the terms in our test document. Thus for every mood the sum of the  $tf-idf$  values of the terms

For happy mood:

$$Total_{happy} = tf-idf_{(t_1)} + tf-idf_{(t_2)} + tf-idf_{(t_3)} + \dots + tf-idf_{(t_n)}$$

Similarly  $Total_{sad}, Total_{romantic}, Total_{angry}$  and  $Total_{spiritual}$  are calculated. The  $Total_{mood}$  value is then divided by the count of the number of moods ( $N_m$ ) i.e., five for this system to calculate the average value.

$$Average_{mood} = \frac{Total_{mood}}{Number\ of\ moods}$$

Thus we get an average value of  $tf-idf$  weights for every mood as  $Average_{happy}, Average_{sad}, Average_{romantic}, Average_{spiritual}$  and  $Average_{angry}$ . These average values are compared and the mood which has the highest average value is assigned to the mood of the test document.

### IV. EXPERIMENTS AND RESULTS

To predict the mood of the song based on  $tf-idf$  values, a Java program was utilized. The experiments were done on the 500 training songs dataset. The system was tested using  $tf-idf$  and  $tf * tf-idf$  values for both the algorithms. The results are given in the Tables 1 and 2.

Table I.  $tf-idf$  weighing scheme algorithm

Song	Happ y	Sa d	Angr y	Spiritua l	Romanti c	Ac c %
Happy	55	0	0	0	49	53
Sad	10	23	0	0	68	23
Angry	8	4	50	1	38	50
Romanti c	1	0	0	0	95	96
Spiritual	1	2	0	71	28	70

Table II.  $tf * tf-idf$  weighing scheme algorithm

Song	Happy	Sad	Angry	Spiritual	Romantic	Acc%
Happy	65	2	0	0	36	64
Sad	16	25	0	0	54	27
Angry	15	3	51	2	28	52
Romantic	5	1	0	0	91	93
Spiritual	7	1	0	66	23	68

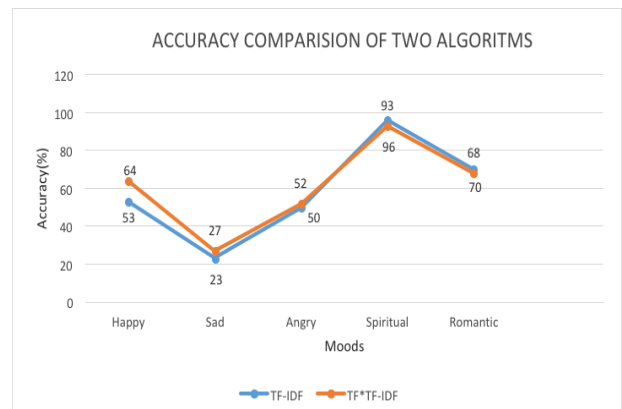


Fig 1. Line graph for algorithm comparison of  $tf-idf$  and  $tf * tf-idf$  in terms of accuracy



Fig.1. gives the good accuracy percentages for the moods happy, angry, spiritual and romantic were observed except for sad mood. The accuracy for romantic and happy are too good, because they consist of a lot of mood specific words such as “happy”, ”good”, ”feel”, ”sunshine”. Similarly for spiritual also accuracy is good using the specific words like “god”, “lord”, “holy” and “Jesus”. The words that hold more importance to the mood has higher value. Comparing the results obtained from the two weighing scheme algorithms, the accuracy percentage obtained for *tf\*tf-idf* is slightly higher than that of the *tf-idf*. For example for happy mood, the accuracy percentage is 53% in *tf-idf* which improves up to 65% in *tf\*tf-idf*. The similar trends were observed for other moods too. Thus, the results show that *tf\*tf-idf* is comparatively better than *tf-idf*. Table 4 contains the confusion matrix of *tf-idf* results; this will help us in determining the performance of the algorithms. The confusion matrix is a very important theoretical concept in predictive analysis. It reports the number of false positives (fp), false negative (f<sub>n</sub>), true positives (t<sub>p</sub>) and true negatives (t<sub>n</sub>). Using these values, Precision, Recall and F-measure can be calculated which are basically measure of relevance. Precision (also called positive predictive value) is the fraction of retrieved instances that are relevant. i.e.,

$$\text{Precision} = \frac{tp}{tp + fp}$$

While recall (also known as sensitivity) is the fraction of relevant instances that are retrieved.

$$\text{Recall} = \frac{tp}{tp + fn}$$

F-measure can be defined as the harmonic mean of precision and recall.

$$F = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}$$

**Table III. Precision, Recall and F-Measure Results of *tf-idf***

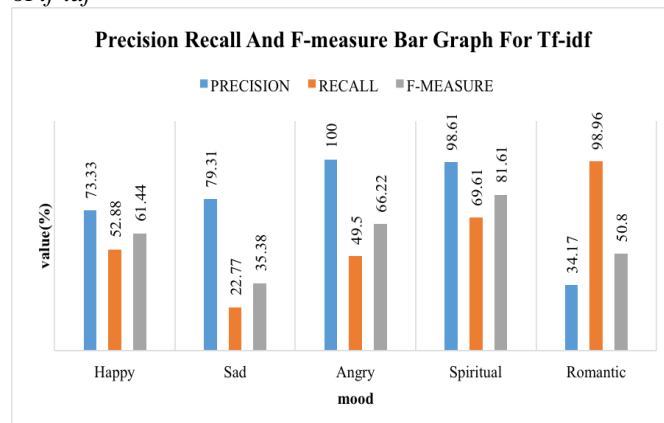
Mood	Precision	Recall	F-Measure
Happy	73.33	52.88	61.44
Sad	79.31	22.77	35.38
Angry	100	49.5	66.22
Spiritual	98.61	69.61	81.61
Romantic	34.17	98.96	50.8

Table III lists Precision, Recall and F-Measure for various moods of the algorithm *tf-idf*. Table IV lists Precision, Recall and F-Measure for various moods of the algorithm *tf\*tf-idf*

**Table IV. Precision, Recall and F-Measure Results of *tf\*tf-idf***

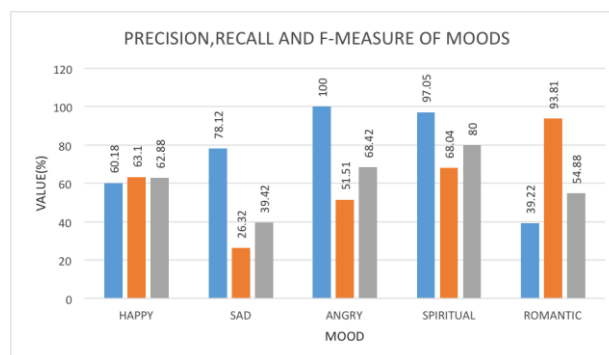
Mood	Precision	Recall	F-Measure
Happy	60.18	63.10	62.88
Sad	78.12	26.32	39.42
Angry	100	51.51	68.42
Spiritual	97.05	68.04	80
Romantic	39.22	93.81	54.88

The precision values for the spiritual and angry moods are considerably very high. This high precision value of the moods can be attributed to the high weights and high frequencies of the relevant words related to the respective moods. Figure 2 shows the Precision, Recall and F-measure of *tf-idf*



**Fig 2. Precision recall and F-Measure bar graph for *tf-idf***

Figure 3 shows the precision, recall and f-measure of *tf\*tf-idf*



**Fig 3. Precision Recall and F-Measure Bar Graph For *tf\*tf-idf***

## V. CONCLUSION

Research works are being carried out to develop methodologies that can distinguish the music based on individual mood. In this research, instead of using the traditional method of audio feature analysis, we had proposed a system which analyses the lyrics dataset of the songs based on the features extracted from the training phase and from which we can predict the mood of the song that is presented to the system at the validation stage. The proposed system had considered five moods such as happy, angry, sad, romantic and spiritual and each containing one hundred songs each, for the Validation purpose. The system is able to predict the mood of the song based on the analysis of the lyric text.

In this paper, the system was trained using the data set, which contained 500 songs. Based on the results obtained from the training phase, the system had predicted the mood of the song in the validation phase. The system was tested with two algorithms which are *tf-idf* and *tf\*tf-idf*. The overall accuracy achieved was about 59% in the case of the *tf-idf* weighing scheme and about



63% through using *tf\*tf-idf* algorithm. In future, the experiment can be extended a large sized dataset of songs in order to train the system better and improve the accuracy of prediction. The proposed method can be extended to implement a system for mood classification for other languages. This paper can be extended to further improve the system and its accuracy by including hybrid features such as audio and lyric text.

## REFERENCES

1. Menno van Zaanen, Pieter Kanters. 11th International Society for Music Information Retrieval Conference (ISMIR 2010).
2. Vipin Kumar, Sonajharia Minz. "Mood Classification of Lyrics using SentiWordNet". International Conference on Computer Communication and Informatics (ICCCI -2013), Jan. 04 – 06, 2013, Coimbatore, INDIA (2013).
3. Teh Chao Ying, Shyamala Doraisamy, Lili Nurliyana Abdullah. 978-1-4673-1090-1/12/\$31.00 ©2012 IEEE.
4. X. Hu, and J.S. Downie, Proceeding of the 2010 Joint International Conference on Digital Libraries, JCDL (2010).
5. Xiao Hu J. Stephen Downie. When Lyrics Outperform Audio For Music Mood Classification: A Feature Analysis. 11th International Society for Music Information Retrieval Conference (ISMIR 2010).
6. X. Hu, J.S. Downie and A.F. Ehmann, Lyric Text Mining In Music Mood Classification, In Proc. Of the 10th International Conference on Music Information Retrieval. (ISMIR'09).
7. R. Mayer, R. Neumayer, and A. Rauber, "Combination of Audio and Lyrics Features for Genre Classification in Digital Audio Collection,"
8. H. He, J. Jin, Y. Xiong, B. Chen, W. Sun, and L. Zhao: "Language Feature Mining for Music Emotion Classification via Supervised Learning From Lyrics," In Proceedings of Advances in the 3rd International Symposium on Computation and Intelligence (ISICA'08).
9. Y. Hu, X. Chen, and D. Yang: "Lyric-Based Song Emotion Detection with Affective Lexicon and Fuzzy Clustering Method," In Proceedings of the 10th International Conference on Music Information Retrieval (ISMIR'09).
10. Teh Chao Ying, Shyamala Doraisamy and Lili Nurliyana Abdullah, Lyrics Based Genre Classification Using Variant *tf-idf* Weighting Schemes. Journal of Applied Sciences, 15: 289294 (2015).
11. Kanchana S, Meenakshi K, Ganapathy V, "Comparison of genre based tamil songs classification using term frequency and inverse document frequency", Research Journal of Pharmacy and Technology Volume 10, Issue 5, 1 May 2017, Pages 1449-1454.
12. Vishnupriya, Meenakshi K "Automatic Music Genre Classification using Convolution Neural Network", 2018 International Conference on Computer Communication and Informatics, ICCCI 2018.T