An Adaptive Approach to Identify Genre in Music Videos Using Word2Vec Model

S. Metilda Florence*  
Research Scholar, Bharathiar University, Coimbatore, Tamil Nadu, India

S. Mohan  
Associate Professor, CCIS, Al Yamamah University, Kingdom of Saudi Arabia

DOI: 10.23956/ijarcsse/V7I7/01707

Abstract—The vector representations of words presented by Word2Vec model have been shown to be very useful in many application developments due to the semantic information they convey. This paper proposes a similar form, the MusicGenre2Vec. MusicGenre2Vec represents the numerical genre features of music segments inside the vector with the intention to describe the phonetic systems of the tune segments in an excellent way. We are hoping the vector representations obtained in this way can describe more precisely the phonetic structures of the Music indicators, so the Music segments that sound alike would have vector representations close by within the space. This form of depiction is called MusicGenre2Vec in this paper. The proposed system gives 80% of accuracy in finding the Genre of a video song.

Keywords—Word2Vec, learning models, Genre similarities, Video Annotation, Semantic information

I. INTRODUCTION

Cisco states that, in 2020, 79% of global Internet traffic will be by Internet videos. The world will reach one million videos viewing every second. From Cisco’s statement, we can say that people watching videos on the Internet will dramatically increase. To help the music lovers to locate the video song of their interest, a novel methodology is implemented in this paper. The proposed system will search the video songs based on their content and not on Meta data. This article focused on the Genre of a video song. Six Genres are taken for analysis, namely Blues, Electronics, Jazz, Pop, Raphi and Rock.

The more contributing Video songs on Genres described above are collected from the Internet. For each Genre 100 songs are taken, totally 600 Video songs are collected for all the six Genres. Music information from the songs is collected as numeric data called features. Rhythmic and Cepstral features are extracted from the songs by applying mathematical models. The novelty of this paper is numerical features are represented as words and applied to Word2Vec model for finding semantic similarities of Genres. Section 2 discusses the methodology of our work. Implementation and Results are presented in Section 3. Section 4 gives the conclusion.

II. METHODOLOGY

In this paper, we have proposed an unsupervised learning of Genre in a Video song using distributed representation of Musical sequences. The music signals are represented as an n-dimensional Vector named as MusicGenre2Vec, and it describes the Genre of a song using neural networks.

A. Video Songs Collection

Based on Blues, Electronics, Jazz, Pop, Raphi and Rock Genres, more contributing video songs are downloaded from the Internet. For each Genre 100 songs are taken, totally 600 songs are collected for all the six Genres. Using video cutter tool these songs are trimmed to 10 seconds duration uniformly.

B. Extraction of Audio Signal and Pre processing

A program written in Matlab is used to extract the audio signals from the video song. Wiener Filter preprocesses the extracted signals. Wiener Filter is used to filtering noise and unwanted signals from the audio signal. It separates signals based on their frequency spectra [1]. This filter blocks noise frequencies and allows only signal frequencies. The Wiener Filtered signal \( W[f] \) is given by

\[
W[f] = \frac{S[f]^2}{S[f]^2 + N[f]^2}
\]

where \( N[f] \) and \( S[f] \) are frequency spectra of the noise and signal respectively.

At the end of the preprocessing the compressed and filtered audio signals are applied to the feature extraction stage.

C. Feature Extraction

Feature extraction is a major phase in supervised and unsupervised learning. We need to identify more contributing features for analysis. Because most intelligent algorithms can also perform badly if noncontributing features are used, while simple algorithms can execute well when they are served with the contributing features. More contributing features are selected in this work, namely Rhythmic and Cepstral features. From Rhythmic features, Rhythm Histogram is taken and from Cepstral features, Mel Frequency Cepstral Coefficient (MFCC) is taken for extraction.
1) Extracting Rhythm Histogram
A Rhythm Histogram (RH) aggregates the modulation amplitude values of the individual critical bands computed in a Rhythm Pattern and is thus a lower-dimensional descriptor for general rhythmic characteristics in a piece of audio [2]. A modulation amplitude spectrum for critical bands is calculated, as for Rhythm Patterns according to the Bark scale. Subsequently, the magnitudes of each modulation frequency bin of all critical bands are summed up to a histogram, exhibiting the magnitude of modulation for 60 modulation frequencies between 0 and 10 Hz. By taking the median of the histograms of every 6 second segment of a given piece of audio, the Rhythm Histogram feature is calculated.

2) Extracting MFCC
The audio signal is divided into a number of overlapped frames to extract MFCC feature. Each frame is multiplied by a Hamming window $hw(h)$ to minimize the ringing effect [3], which is shown below:

$$ hw(h) = 0.54 - 0.46\cos\left(\frac{2\Pi h}{N-1}\right), 0 \leq h \leq N - 1 $$

where $N$ is the size of the Hamming window. To obtain the spectrum for each Hamming windowed frame, Fast Fourier Transform (FFT) is applied. The audio models are sampled at 44.1 KHz. To get an accurate FFT, music samples are segmented into 23 ms frames [4]. When compared with other sample rates and segment size combinations, this gives the best performance [5]. For windowing, Hamming window is used because the combination of Mel frequency and Hamming window gives improved results [6]. Thirteen MFCC coefficients are calculated for each window. To reduce the size of the feature set, the mean and standard deviations of features are obtained.

D. Word2Vec Model
Distributed representation is one of the most efficient methods in machine learning [7]. The principal aim of this method is programming and storing information about an item within a system through creating its relationship with other items. In human memory, the items are stored in a “content-addressable” manner. These kinds of storage help us to recollect the items from the partial description. The content addressable items and their properties are stored within a closed region give a feasible structure to generalize features attributed to an item.

In training step, a distributed representation breaks the sequences into sub sequences. The simplest and most common technique in study sequences involves a fixed-length overlapping of n-grams [9-11]. We utilize n-gram modeling for training a general purpose distributed representation of Music Genre sequences. There are two n-gram models are available in Word2Vec namely Continuous Bag of Words (CBOW) and Skip-gram.

The general meaning of CBOW and Skip-gram are explained from Jono's example below:
The sentence “Hi Iylin how was the Sweet?” becomes:

- Continuous bag of words (3-grams):
  ```
  {"Hi Iylin how", "Iylin how was", "how was the", ...}
  ```

- Skip-gram (1-skip 3-grams):
  ```
  {"Hi Iylin how", "Iylin how was", "Iylin how the", ...}
  ```

Notice "Hi Iylin was" skips over "how".

Since Skip-gram works well with the smaller number of training data [12] and represents well even rare words or phrases, we have selected Skip gram for our task. Skip-gram with “1 skip, bi-grams” is used in this work.

Exciting patterns have been seen by training word vectors using Skip-gram in natural language. Words with related vector representations show multiple degrees of relationships [8]. For instance, “Man is to Woman, Boy is to?” The Vector $\text{Man} - \text{Boy} + \text{Woman}$ resembles the closest vector to the word $\text{Girl}$.

In this work, unique patterns in music sequences are sought to facilitate Genre classification. How Skip-gram can be used to train a Distributed representation for Musical sequences over a large set of sequences, and establish the Genre of a song for such representations are presented here.

A novel algorithm MusicGenre2Vec is proposed to represent the musical information in the form of words. Proposed algorithm is given below.

Algorithm: MusicGenre2Vec
Step 1: Extract Cepstral and Rhythmic features from each audio track by applying mathematical models and create an FEATURESET.CSV file.
Step 2: For each ROW in FEATURESET.CSV repeat step 3 to 6
Step 3: $\text{FINAL_STRING} = '<' + \text{Genre} + ' ' + \text{ROW} + ' >$
Step 4: For $I = 1$ to Length(ROW) repeat step 5
Step 5: $\text{FINAL_STRING} += \text{pair}[I-1] + ' ' + \text{pair}[I] + ' <' + \text{Genre} + ' >$
Step 6: Write $\text{FINAL_STRING}$ into Processed data file.
Step 7: Train the Processed Data set with Skip gram Model.
Step 8: Apply Negative Sampling to get the accurate context

© www.ijarcsse.com, All Rights Reserved
Florence et al., International Journal of Advanced Research in Computer Science and Software Engineering 7(8)
ISSN(E): 2277-128X, ISSN(P): 2277-6451, DOI: 10.23956/ijarcsse/V7I7/01707, pp. 20-24

In Step 1, extracted audio features are stored in FEATURESET.CSV file.

In Step 2, each ROW from the FEATURESET file is taken and processed in Step 3 to 6.

In Step 3, FINAL_STRING is initialized to Genre.

For example, FINAL_STRING = <Rock>

In Step 4 and 5, each numerical values in the ROW is converted into sentences of words, and stored in FINAL_STRING variable.

For instance,

FINAL_STRING = <Rock> 7.774 3.543 <Rock> 2.641 2.864 <Rock> 2.125 1.785 <Rock> 2.099 1.438 <Rock> 1.721 1.883 <Rock> 1.406 1.755 <Rock> 1.94 0.001587 <Rock> 0.2232 0.09545 <Rock> 0.09081 0.09411 <Rock> 0.09184 0.08605 <Rock> 0.07749 0.06888 <Rock> 0 -232.9 <Rock> 38.7 9.704 <Rock> -8.943 -1.877 <Rock> 3.994 2.22 <Rock> -1.411 0.3005 <Rock> 2.394 0.8221 <Rock> -1.888 -1.732 <Rock> -0.9971 0.6332 <Rock>

In Step 6, FINAL_STRING will be stored into Processed data file.

In Step 7, the Processed data is applied into Skip gram neural network model for training and finally, in Step 8, Negative Sampling is applied to get the accurate context.

By applying the MusicGenre2Vec algorithm on the extracted features, a list of shifted non-overlapping words instead of taking overlapping windows is generated, as shown in Fig. 1.

The next step is, training the shifted non-overlapping data through a Skip-gram neural network. For a given training sequence of words l_1, l_2, l_3...l_N we would like to find their corresponding n-dimensional vectors maximizing the following average log probability function given in (3). Such a constraint allows similar words to assume a similar representation in this space.

![Fig. 1 A portion of shifted non-overlapping words](image)

where d is the size of the training context (which can be a function of the center word l_i). The higher value of d results in more training examples and thus can lead to a higher accuracy, at the cost of the training time. The basic Skip-gram design defines the softmax function p(l_i+j | l_i).

\[
p(l_i+j | l_i) = \frac{\exp(v^T_{l_i+j}v_{l_i})}{\sum_{k=1}^{L} \exp(v^T_{l_i+k}v_{l_i})}
\]

(4)

where N is the size of the training sequence, 2d is the window size we consider the context, l_i is the center of the window, L is the number of words in the dictionary collection of words and v_i and v_i are input and output n-dimensional representations of word l, respectively. The probability p(l_i | l_i) is defined using a softmax function. An efficient approximation of a softmax function is Hierarchical softmax or negative sampling. In our implementation, negative sampling is used [13], which is considered as the advanced method for training word vector representation. The negative sampling is evaluated using the following expression.

\[
\log \sigma(v^T_{l_i+j}v_{l_i}) + \sum_{i=1}^{k} E_{l_{-i}p_{-i}(i)}[\log \sigma(-v^T_{l_i}v_{l_{i+j}})]
\]

(5)
The negative sampling is used to replace every \( \log p(l_{i+j} | l_i) \) in (3). The aim here is to differentiate the word \( l_{i+j} \) from the noise distribution \( p_n(l) \) using logistic regression, where there are \( k \) negative samples for each data sample. For small dataset, the \( k \) values range from 5 – 20, and for large datasets the \( k \) range can be from 2 – 5.

III. EXPERIMENTS AND RESULTS

Gensim is one of the best frameworks that efficiently implement algorithms for statistical analysis. This model is used to compute the semantic similarity between words using the mathematical vector representation. Gensim uses NumPy and SciPy for good performance. NumPy provides support for large and multidimensional arrays and set of mathematical functions to work on it. It is an extension to the Python programming language. For scientific computing and technical computing SciPy is used. SciPy is an open source Python library which contain modules for optimization, linear algebra, integration, Fast Fourier Transform and so on. Gensim can run on Windows, Linux and any other operating system which supports NumPy and Python. Gensim relies on NumPy, SciPy and Python software. First, we need to install Python, then SciPy and NumPy, finally Gensim is installed.

Our objective is to construct a distributed representation of Musical systems. In the training stage, a large amount of features are applied to the training algorithm to confirm that sufficient conditions are perceived. For each Genre 100 video songs are collected from the Internet. We have created a database with 600 video songs. All the video files are manually trimmed to 10 seconds duration. Audio signals are extracted from video and preprocessed. Mathematical models are applied to preprocessed signals in order to extract most contributing features namely Rhythmic and Cepstral features. Skip-gram neural network model is used to train the dataset. Negative sampling is applied on the softmax function to get the accurate results. Fig. 2 shows the running environment of the Genre identification system. One hot vector is given as an input, and the result got as any of the aforesaid Genres. From each Genre, 60 songs are taken for training in Skip-gram neural network model and remaining 40 songs are used for testing. For instance, in Blues Genre 60 songs are trained and remaining 40 songs are used for test the model. Out of 40 songs, the implemented system predicts 30 songs correctly. Similarly, the same procedure is applied to remaining Genres. The proposed methodology gives the maximum of 80% accuracy in identifying the Genre. Fig. 3 depicts the result analysis of Genre identification system.

In our previous work, using Multiclass classifier and SMO classifier algorithms, we achieved maximum of 73 % and 72% accuracy [14]. Our proposed MusicGenre2Vec algorithm gives 80% of accuracy in identifying the Genre of the video song. The performance analysis of the proposed MusicGenre2Vec with Multiclass and SMO algorithms are depicted in Fig.4.
A novel and efficient approach for identifying a Genre in the video song is presented. This methodology enables the music lovers to choose their favourite video song. In this Music Genre Identification system, six Genres namely Blues, Electronics, Jazz, Pop, Raphi and Rock are selected for analysis. For each Genre, 100 video songs of length 10 seconds duration are considered. From these video songs the audio signals are extracted. Mathematical functions are applied to calculate the Rhythmic and Cepstral features from the extracted signal. These numerical features are converted to word by using a novel algorithm called MusicGenre2Vec. Skip-gram model is used to train the dataset, and negative sampling is used to find the accurate result. MusicGenre2Vec algorithm gives a maximum of 80% accuracy in identifying a Genre of a video song. This Framework restricts the identification for 10 seconds period for experimental purpose. This duration can be extended for the entire song. In future, this work can be extended to cover more types of Genres. Size of the feature set can also be increased by including more contributing features from the audio signal. The increase in features may increase the accuracy percentage.

REFERENCES