Classification of Music Genre with Long Short-Term Memory (LSTM)

Bharat Chawla
Khoury College of Computer Sciences
Northeastern University
Boston, Massachusetts
chawla.bh@northeastern.edu

Himaja R. Ginkala
Khoury College of Computer Sciences
Northeastern University
Boston, Massachusetts
ginkala.h@northeastern.edu

***Abstract*—The classification of music genres, particularly when relying solely on lyrical data, continues to pose a challenge for the Music Information Retrieval industry. Streamlining the music classification process could enhance the efficiency of recommending new songs to the users. To address this issue, this study explores a type of recurrent neural network model called LSTM with attention mechanisms to learn the importance of words, lines, and segments. The model was tested on lyrical data of 20,000 songs from the top 10 English language genres, and experimental results with the primary model showcased the desired efficacy. As a result, the model provided useful insights from a computational perspective to lyrical structure features that differentiate musical genres.**

***Keywords*—classification, music genre, neural network, NLP, LSTM, RNN**

I. Introduction

Music genre entails the classification of music into distinct groups categorized by shared traditions, origins, or distinctive styles. Traditionally, a human expert would conduct these classification processes, evaluating a song's resemblance to a specific genre manually. Though, due to the subjective nature of music, this organization process is not a straightforward task; a song perhaps belongs to multiple musical genres, and these genres are continuously evolving.

In an era where music has been seamlessly integrated into technology, companies such as Spotify and Apple Music have shaped the way music is created, personalized, and experienced. Out of all the music streaming companies, Spotify emerges as the dominant force, with an impressive user base exceeding 500 million monthly active listeners. What motivates this enormous user base isn't just the platform's vast music collection, but the ingenious integration of AI-powered recommendations tools which are capable of creating a personalized musical journey for every distinct user. A key component for such platforms are the recommender systems that suggest tracks, albums, and artists tailored to an individual user's specific needs and preferences. Therefore, labeling songs into categories of genres is the first step towards user catered playlists.

Presently, the majority of streaming platforms rely on metadata, encompassing elements like acoustics and emotional tone for genre classification. Given the intricacies associated with extensive lyrical datasets, many companies opt to overlook lyrics in the context of classification problems. However, it is hypothesized that utilizing lyrical data could prove to be beneficial in music genre classification. Human inclination may be to consider audio files more valuable than lyrical data for genre classification. However, given the higher dimensionality of audio data and the lower dimensionality of lyrics, the goal is to evaluate how effectively and accurately songs can be categorized into genres based solely on their lyrical content.

Most of the current methods involve classifying genres using musical features as well as feeding lyrics into neural networks. However, their shortcomings include failing to take word sequences into consideration. In an effort to make up for this deficiency, this study aims to leverage a specific type of recurrent neural networks, known as Long Short-Term Memory (LSTM) networks, in addition with the attention mechanism and GloVe word embeddings to effectively classify music genres.

II. Related Work

A. A Novel Approach to Music Genre Classification using Natural Language Processing and Spark

Recommendation systems are an extremely important component of music streaming services such as Spotify and Apple Music as they work to cater as accurately as possible to a user's musical preferences. To sustain these systems, effective labeling data to identify genres become vital. Recent work in music genre classification has predominantly explored deep neural networks, however S. Duggirala and T.-S. Moh aimed to employ advanced techniques such as Hierarchical Attention Networks (HANs) from text classification. The process began with the input of an audio file which is separated into distinct components (drums, vocals, bass, and accompaniment), converted to symbolic text data, and then appended into a single text representation. This representation is then converted into a word embedding representation that is fed as input to the HAN. Different combinations of genres were tested together and the approach was met with sophisticated results: {pop, rock, jazz, rap} scored the highest with an F1 score of 0.92. S. Duggirala and T.-S. Moh noted multi-label classification as an avenue of research for the future as more and more artists are deriving inspiration from various genres and working to combine them in creative ways.

B. Music Genre Classification: A N-Gram Based Musicological Approach

E. Zheng, M. Moh, and T.-S. Moh took on a musicological approach to genre classification. A dataset comprising 200 records of classical jazz, and ragtime music was utilized, specifically focusing on symbolic piano works. Post preprocessing, n-gram count vectors were used to produce input needed for the classification model. The primary classifier was chosen to be Multinomial Naive Bayes as it is a classic probabilistic classifier for text classification tasks. The experiment resulted in high prediction accuracies averaging at 90% and peaks of 98%. E. Zheng, M. Moh, and T.-S. Moh concluded that there is scope for more research in this specific application that could be explored via larger, more comprehensive, and various instrumental music datasets.

III. Data

The classification of songs into musical genres such as Pop, Rock, Country, etc. will be accomplished through the analysis of a repository of data sourced from Kaggle.

A. Repository

The sourced dataset includes two subdatasets, one of artists data and another of lyrics data. The artists dataset consists of 4,168 records and is comprised of the following columns: “Artist”, “Genres”, “Songs”, “Popularity”, and “Link”. The lyrics dataset consists of 383,570 records and includes the following columns: “ALink”, “SName”, “SLink”, “Lyric”, and “Language”.

B. Prepossessing

Since the original dataset was divided into two separate subdatasets, they needed to be merged in a way that did not comprise either dataset’s content. Despite their distinct contents, both datasets shared a common column for an artist’s link, “Link” in the artists dataset and “ALink” in the lyrics dataset. This facilitated their easy integration into a singleton dataframe. This merge showcased 4,161 unique artists, 79 genres, 266,634 songs, and 52 languages in the data. While the initial dataset encompassed lyrics in a plethora of languages, this study exclusively focuses on the English language. Therefore, after filtering the data to show just English records, the dataset was narrowed down to 2488 unique artists, 73 genres, and 137932 songs.

Further data exploration uncovered that the lyrics contained links, punctuation, stopwords, and various non-alphanumeric characters, which were deemed irrelevant for predictive purposes. To tackle this issue, a refined version of the lyrics was created. It was determined that the classification task would be with the top 10 genres in the English language; the frequency of songs present in these genres can be seen in Figure 1 below. Upon closer examination, it became evident that the dataset was imbalanced. Such datasets can lead the model to be biased toward the majority class, result in suboptimal performance, and may hinder the model’s ability to generalize well to new, unseen data. Therefore to preemptively tackle these issues, a balanced dataset was created by randomly selecting 2,000 songs from each genre, resulting in a total of 20,000 songs for training and testing.

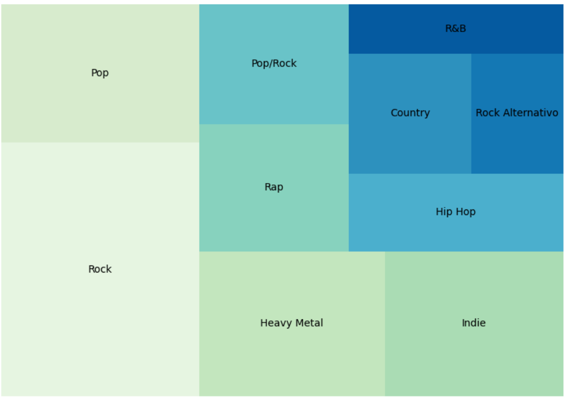

Figure 1. Top 10 English Genres Frequency Treemap

IV. Methodology

A. Long Short-Term Memory (LSTM)

Long Short-Term Memory (LSTM) networks represent a distinctive category within Recurrent Neural Networks (RNNs), and have proven to be effective when handling extended-term data sequences. They represent an advancement over traditional RNNs, mitigating the challenge of vanishing gradients commonly associated with them. The vanishing gradient problem occurs when training deep neural networks, particularly those with recurrent connections as it becomes difficult for the model to remember and learn long-term sequences in the data being trained on.

To address the vanishing gradient problem, *Hochreiter et al*. [5] proposed a solution that incorporates special memory cells and gating mechanisms into their state dynamics. Each cell state in LSTM is accountable for a decision that remembers or forgets information. Figure. 1 shows the general architecture structure of the LSTM network. It consists of three gates: input gates, forget gates, and output gates. The purpose of these gates is to retain and update the status of the memory unit.

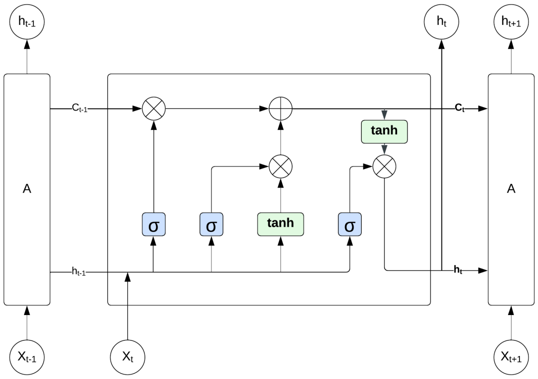

Figure 2. Module Structure of LSTM

There are three ways in which the memory unit maintains status: (1) The forget gate determines which cell's memory should be dropped from the memory unit based on error backpropagation. (2) The input gate decides which new information should be stored in the memory unit. This operation consists of two parts: first, a sigmoid layer decides the extent to which a new value flows into the cell. Afterwards, the tanh layer creates a vector of new candidate values. (3) The output gate decides what should be output to the next cell as a hidden vector. This action also consists of two parts similar to the input gate.

Although LSTMs exhibit proficiency in capturing temporal dependencies within sequence, they struggle when confronted with long-term dependencies. To overcome these dependencies, *Bahdanau et al.* [6] proposed a computational mechanism known as the attention which focuses on different parts of input data when making predictions or decisions. The mechanism, visualized in Figure 3, was introduced to generate deep features containing the attention weights. It is frequently applied in neural text translation, particularly where a context vector *C* is influenced by a target sequence *y*. This medium can highlight the effect of input on the output by giving more weight to the relevant information and reducing the side-effects of less relevant parts.

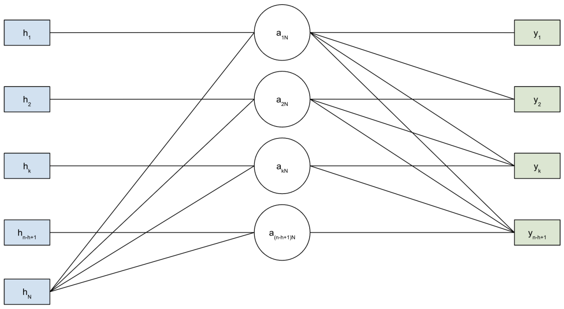

Figure 3. The Attention Mechanism

B. Cross-validation

One commonly used statistical method used to estimate the skill of machine learning models is k-fold cross validation. This procedure is used to estimate the skill of a model on new data. For example, a 10-fold cross validation divides a dataset into 10 randomly-selected folds. During a run, each fold has a turn as the test set, while the other folds together become the training or validation sets. This process is repeated 10 times, such that each fold has a turn as the test set. The average error of all the runs is the total error. This technique prevents overfitting and additionally prevents any variance in the sampling from adversely affecting the model.

C. Ablation

The removal of a component from a learning system is known as ablation. Hence an ablation study investigates the performance of a system by removing certain components in order to understand the contribution of those particular components to the overall system. In the context of a machine learning model, methods of exploration include removal of particular variables, removal of parts of the original model, downsampling the data set, etc.

V. Experiments and Results

A. Primary Model

The main model constructed for this classification problem consists of the following neural network layers: (1) input layer, (2) embedding layer of GloVe word embeddings, (3) LSTM layer, (4) attention layer, (5) global average pooling layer, (6) dropout layer, and (7) linear layer. A batch size of 100 was used with a dropout rate of 0.2, and the model was trained for 15 epochs.

The training and validation metrics of the model are showcased in Figure 4. A training accuracy of 88.23% was observed while the validation accuracy was 34.36%. Similarly, it can be seen that the score of loss of training reduces, as expected, but increases for the validation set. This situation of high training scores for accuracy and loss but low ones of the same for the validation set is indicative of the model succumbing to overfitting. Both the precision and recall values for training were increasing as desired of an effective model, however the values for the validation set were declining for precision, and only slightly increasing for recall. This observation once again points to the possibility that the model is likely overfitting to the training data and is not generalizing well to new, unseen data.

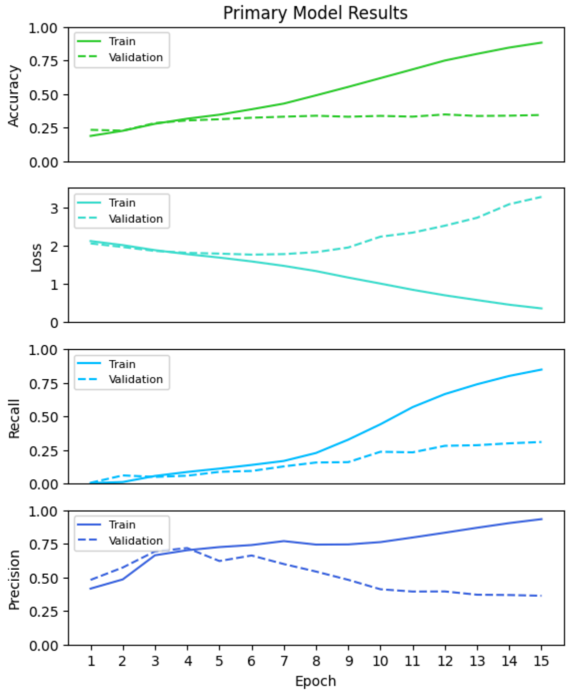

Figure 4. Primary Model Metrics

B. Cross-validation

A 5-fold cross-validation was performed on the primary model to evaluate its performance, analyzing metrics such as training accuracy, loss, recall and precision as illustrated in Figure 5. The cross-validation study resulted in slight changes for training accuracy across each fold, ranging from 56% to 57%.

Similarly, the cross-validation exhibited fluctuation in training loss across each fold, varying from 1.07 to 1.17. Variations in loss could be due to the dataset's diversity, where each fold may introduce a slightly different data distribution. Moreover, models such as LSTM are known for their sensitivity to variations in training data since the model experiences different subsets in each fold, and therefore may potentially result in performance fluctuations.

Additionally, the cross-validation study showcased recall and precision values which demonstrated fluctuations across each fold ranging from 0.70 to 0.80 for precision and 0.38 to 0.36 for recall. Potential reasoning for performance fluctuations may be attributed to the model's sensitivity to variations in training data, given its exposure to distinct subsets in each fold.

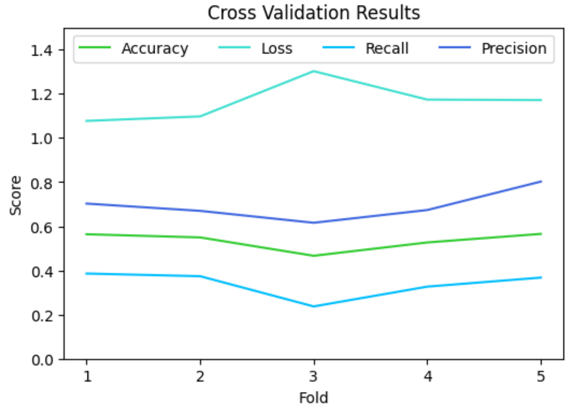

Figure 5. Cross Validation Training Metrics

C. Ablation

The primary model was manipulated in different ways to produce three distinct models for an ablation study: (1) elimination of global averaging pooling layer, (2) removal of both the attention layer and global averaging pooling layer, and (3) reduction of the size of hidden layer dimensions.

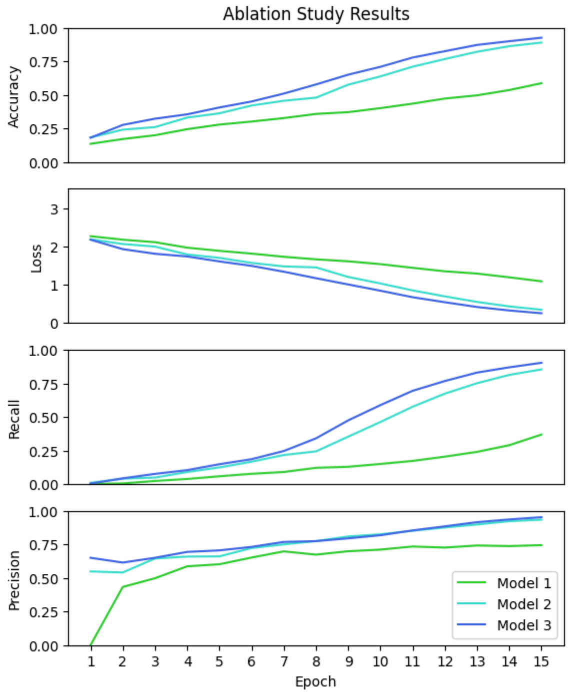

Figure 6. Ablation Study Training Metrics

Firstly, the removal of the global average pooling layer is done to visualize the effect of this change on the model's performance. The idea of global average pooling is to capture the average intensity or activation of features across the entire domain. It was added to the primary model in efforts to control overfitting and improve the model's ability to generalize to different inputs. The removal of this layer was experimented with in the ablation study to see how deeply its absence would affect the model, and as can be seen in Figure 6, Model 1 shows a significant decrease in training accuracy score, reaching a maximum of 58.70% over 15 epoch cycles. The confusion matrix of the test set for Model 1 can be seen below in Figure 7, resulting in an overall testing accuracy of 32.95%.

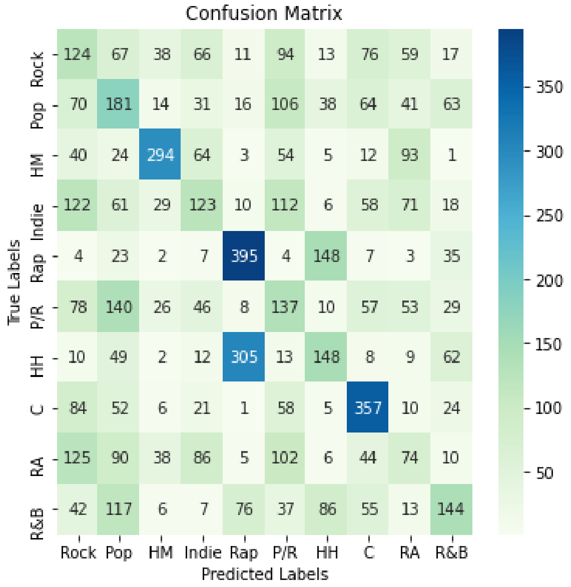

Figure 7. Confusion Matrix of Model 1

Secondly, both the attention mechanism and global average pooling layers were omitted from the primary model to produce Model 2 of the ablation study. Through Model 1 it was already observed that the global average pooling layer added benefit to the primary model, therefore it was worth experimenting with the absence of both the attention mechanism and global average pooling at the same time to see the effect on the primary model. Interestingly enough, the removal of these two layers together produced a training accuracy of 89.02%, a slight improvement on the primary model's training accuracy of 88.23%. The confusion matrix for this model can be seen in Figure 8, and a testing accuracy of 32.73% was observed.

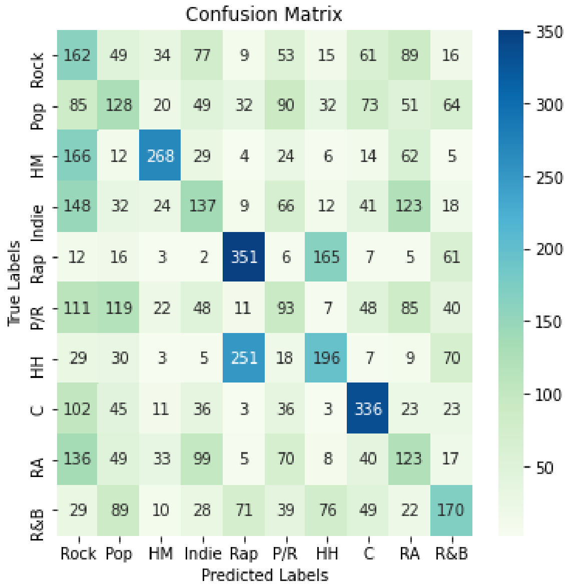

Figure 8. Confusion Matrix of Model 2

The third model of the ablation study was identical to the primary model, but with simply a smaller hidden layer. The original dimension of 128 for the hidden layer was reduced to 64, and the effects observed were quite surprising. Model 3 resulted in a 92.61% training accuracy, outperforming the primary model by more than 4%. However, the testing accuracy of the model stayed similar to those of the previous two models in the ablation study, 33%.

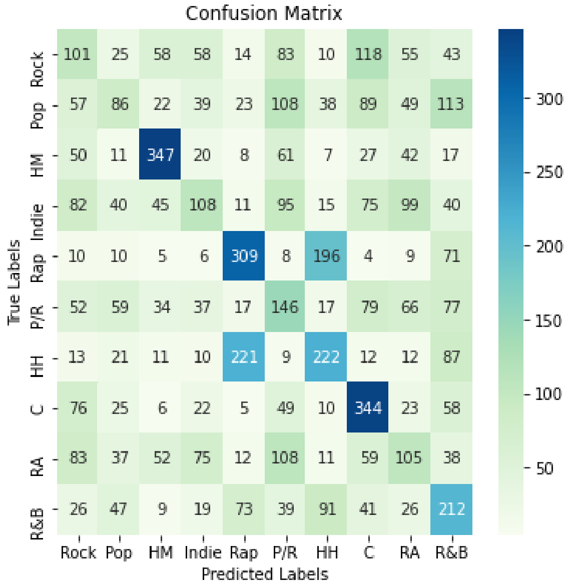

Figure 9. Confusion Matrix of Model 3

Overall, Model 3 demonstrated superior accuracy in the ablation study. This is the scenario where the manipulated model was identical to the primary model, except with a smaller hidden layer dimension. The positive effect on the model's training results could have happened because of a few factors. Firstly, a lower-dimensional hidden layer acts as a form of regularization, and hence helps in preventing overfitting by reducing the model's capacity to memorize noise in the training data. Additionally, a smaller hidden layer promotes better generalization to unseen data by encouraging the LSTM to focus more on essential features and patterns in the data.

VI. Discussion and Conclusions

During initial experimentation, various sizes of datasets were trained to test the maximum capacity of the available machine, as well as the effect of having an imbalanced dataset. Additionally, different sizes of dimensions for the embedding and hidden layer were experimented with for the same purpose of testing the limits of computational power. Therefore the configurations that were utilized in the primary model are those which were settled upon after much trial and error.

Of the models experimented with, a modification to the primary model, a smaller dimension of the hidden layer, proved to result in the best training accuracy. The combination of a LSTM network with the attention mechanism certainly proved to show promise in this particular task of music genre classification. However, reflecting on the low rates of validation and testing accuracies indicated that the model was succumbing to overfitting. Due to limitations of computational power, the model could only be trained on a small portion of the data for which there was access. Therefore while the dataset was balanced and a layer such as global average pooling was included, the problem of overfitting persisted.

In conclusion, this study brought some valuable learnings such as the importance of a balanced dataset, the significance of the smallest changes in models which result in large changes in computational time, and the sheer advantage that having a larger dataset brings. Any future work for this classification problem would benefit from additional computational resources in order to utilize a larger portion of the lyrical dataset to train for a greater number of epochs on the LSTM and attention mechanism network.

References

[1] S. Duggirala and T.-S. Moh, "A novel approach to music genre classification using natural language processing and Spark," *2020 14th International Conference on Ubiquitous Information Management and Communication (IMCOM)*, 2020. doi:10.1109/imcom48794.2020.9001675

[2] E. Zheng, M. Moh, and T.-S. Moh, "Music genre classification: A N-gram based musicological approach," *2017 IEEE 7th International Advance Computing Conference (IACC)*, 2017. doi:10.1109/iacc.2017.0141

[3] Tsaptsinos, A. (2017, July 15). *Lyrics-based music genre classification using a hierarchical attention network*. arXiv.org. https://arxiv.org/abs/1707.04678

[4] Music genre classification using song lyrics - stanford university.(n.d.). https://web.stanford.edu/class/archive/cs/cs224n/cs224n.1214/reports/final_reports/report003.pdf

[5] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," Neural computation, vol. 9, no. 8, pp. 1735–1780, 1997.

[6] D. Bahdanau, K. Cho, and Y. Bengio, "Neural Machine Translation by Jointly Learning to Align and Translate," arXiv preprint arXiv:1409.0473, 2014