Detecting Music Genre from Song Lyrics

Anastasija Mensikova*, Andy Truman*, Shritama Sengupta*, Yunke Zhao*,
*Department of Computer Science
George Washington University

***Abstract*—The world of unstructured textual data is vast, and the need for exploring it and making sense of it is growing rapidly. The purpose of this project is to use various Natural Language Processing (NLP) tools to explore the Lyrics Dataset [2] and use lyrics as means of classifying a song's genre. Although a challenging task, some success was achieved with the means of an LSTM network and pre-trained word embeddings.**

Keywords*—*Machine Learning; NLP; LSTM; Lyrics

I. Introduction

There is a multitude of incredibly interesting Machine Learning problems to explore, some of which are socially critical and some tend to carry more entertainment purposes than any serious intentions. Selecting a problem for this group project appeared quite a challenge for all of us - it is indeed hard to select an interesting and challenging topic out of a sea of varying options. However, we chose to select a problem that resonates with all of us, excites us, yet challenges us to explore subjects and Machine Learning techniques we do not know much about. Performing supervised Machine Learning to classify song lyrics by the genre was exactly a problem that ticked all of those boxes. Although there isn't much innovation in this specific problem, and the presence of the dataset on Kaggle alone indicates its popularity and copious contributions to it from the community, but it is a problem that seemed both interesting, quite unusual from our experiences, and decently difficult.

II. Problem Statement

The main idea of the Machine Learning problem we are exploring is, given song lyrics (in text form), being able to select a music genre that is most closely associated with them. Music has been (and still is) explored on various levels, but for the sake of this project we chose to focus specifically on the lyrics as we want to gain more experience in Natural Language Processing (NLP). Classifying a song's genre is something that all music streaming services are always concerned about. Although there is little knowledge on the exact methodology on how that is performed in the industry, it is safe to assume that a combination of study of both the lyrics and the melody is used to identify the genre to later on deliver that information to the end user (read: us). Although at first this problem might appear to be a trivial case of NLP classification, upon close examination it appears to be far from that. Most NLP problems have a goal of computerising a task that could easily be performed by a human, but for the sake of efficiency it is wiser to make a computer perform that same task. For example, by reading a movie review we can more-or-less get an idea of whether or not it's positive or negative. However, going through film reviews one by one is extremely time-consuming. Song lyrics, however, appear to be much more cryptic. Although it's certainly true that specific lyrics can pertain to specific genres (for example, dark and dramatic stories are not very common in country music, unless it's Taylor Swift of course), it is extremely hard for a human to determine a song's genre purely based on lyrics, unless they are a music expert. In this case, the task is not very intuitive to most human-beings and simply cannot be performed very successfully by anyone. This is exactly what makes this problem so special and us eager to explore it.

III. Related Work

The use of Natural language processing first started approximately in the 1950s, although some work can be found from even former periods. Alan Turing, in 1950, published an article by the title "Computer Machinery and Intelligence" which proposed the idea of Turing Test as an intelligence criterion. Natural Language processing has some a long way and a significant amount of work has been done to find the potential use cases of Natural Language Processing.

Mahedero et al. in their study titled "Natural language processing of lyrics", wrote about an experiment conducted on the use of standard natural language processing (NLP) tools for the purpose of analysing music lyrics since a significant amount of music audio have lyrics that encode an important part of the semantics of a song, complementing their analysis to that of acoustic and cultural metadata and is fundamental for the development of complete music information retrieval systems. A textual analysis of a song can generate ground truth data that can be used to validate results from purely acoustic methods. Preliminary results on language identification, structure extraction, categorization and similarity searches suggests that a lot of profit can be gained from the analysis of lyrics.[4]

Similarly Alexandros Tsaptsinos, in his article "Lyrics-Based Music Genre Classification Using a Hierarchical Attention Network" wrote about his study on music genre classification, especially using lyrics alone, which remains a challenging topic in Music Information Retrieval. This study applied recurrent neural network models to classify a large dataset of intact song lyrics. As lyrics exhibit a hierarchical layer structure—in which words combine to form lines, lines form segments, and segments form a complete song— the researcher adapted a hierarchical attention network (HAN) to exploit these layers and in addition learn the importance of the words, lines, and segments. They finally test the model over a 117-genre dataset and a reduced 20-genre dataset. The study shows that the HAN outperformed both non-neural models and simpler neural models using experimental methods, whilst also classifying over a higher number of genres than previous research. Through the learning process it also visualized which words or lines in a song the model believed are important to classifying the genre. As a result the HAN provided insights, from a computational perspective, into lyrical structure and language features that differentiate musical genres.[5]

Fell et al., in their paper, presented 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. The researchers showed that these can be combined with n-gram features to obtain performance gains on three different classification tasks: genre detection, distinguishing the best and the worst songs, and determining the approximate publication time of a song.[1]

IV. Dataset

We are using a dataset with 250,000+ lyrics over 2k singers from Kaggle [2]. We believe this dataset provides us with enough data to train our model. The dataset is currently the largest dataset related to music and lyrics, containing over 250,000 music lyrics, over 2000 singers, and over 200 genre labels. The songs in this dataset ranged from 1962 to current, and are well-labeled with genre which is be used as our outcome label in training. The main feature we use to train is the lyrics, we have applied several steps to the lyrics to clean it up and make it available for training.

The Kaggle music dataset has over 200,000 songs along with associated lyrics and other metadata. The first task in preprocessing was to reduce the size of the dataset. In order to do this, we first looked at only the top 4 most common genres in the dataset. These were the ‘pop’, ‘rock’, ‘country’, and ‘hiphoprap’ (Hip-Hop/Rap) respectively. Lastly, the lyrics were cleaned of capitalization, digits, punctuation, stopwords, and stemming.

In order to find the top 4 genres in the dataset, we looked at the count of unique songs in each genre and filter out the 4 most popular ones. Each of the four genres had around 110,000 songs at most to 15,000 songs at least. Therefore, we randomly sampled out approximately 15,000 songs from each of the 4 genres mentioned above and created a separate dataset to work on with our model.

In the final step of pre-processing, the lyrical data is cleaned. To accomplish this, we applied each of the cleaning techniques to our newly formed dataset. The first step in our cleaning was to convert all lyrics to lower case. This is to ensure that we do not misclassify proper and improper nouns or consider them as unique words. Then, we removed stop words. Stop words are generally not helpful in determining the overall intent or meaning of the song and only serve to increase the size of the lyrics. Stop words can be words like: ‘the’, ‘a’, ‘and’, etc. Finally, we removed stemming from the words. Stemming may prohibit root words from being detected. Using a common stemmer, like the Snowball Stemmer, allowed us to remove any stems that may have been in our dataset. This way, our learning model only had to process root words.

V. Approach

For the sake of this project we chose Keras as the library of choice. Although there are so many various libraries that perform similar tasks, Keras appears to be the most modular, extensible, and user-friendly one with plenty of useful documentation. Keras provides an easy way to implement a sequential machine learning model. This basically means that each layer in a model has to occur in the order in which it is implemented. Our model contains 4 layers: an embedding layer, a dropout layer, an LSTM layer, and an output layer. Keras is also fits with Jupyter really well, which certainly is a big bonus.

We used the Long Short Term Memory (LSTM) algorithm to build our deep learning model. LSTM is an artificial recurrent neural network structure in deep learning, which is popularly used in NLP tasks. LSTM allows us to save and detect the relationship between different words in a sentence. This makes LSTM a great candidate for recognizing intent or meaning in a body of text. Our approach is relying on the assumption that genre is related to intent or meaning in lyrics, so LSTM is our learner of choice.

The embedding layer is used to embed and feed our data into the model. We embedded lyrics by using Gensim Continuous Skipgram Embeddings, and used a pretrained model to encode preprocessed words in lyrics into Word2Vector type vectors, and then output the result to the next layer.

The second layer is a dropout layer, which, using Spartial-Dropout1D provided by Keras, will randomly dropout entire features from the input to prevent overfitting.

The third layer is the core layer of our model - the LSTM layer. We Created an LSTM layer with 10 hidden units and will randomly drop out 20% of the units for each epoch, furthermore, we will dropout 20% of the recurrent units each epoch.

The last layer is a fully connected (dense) layer with 4 units that represent the output of the model. We used softmax as our activation function, and the output will be a four 0/1 number array that indicate the genre of the given input.

We picked categorical cross entropy as the loss function of the model, and trained the model for 15 epoch. However, early stopping was also implemented as part of the network to ensure a lesser magnitude of possible overfitting and pointless training. The early stopping, the patience of which was set to 3 epochs, allowed the training to stop whenever the validation loss stopped decreasing in 3 consecutive epochs. Although many training experiments were run to test out various possibilities of the network, the latest structure, set to train for 15 epochs, only trained for 7 epochs.

VI. Experiments

A lot of experiments were performed to ensure that the best model infrastructure was used for training. Although, of course, not all possible models were experimented with, some variations included changing the optimizer from adam to adadelta and vice versa, as well as changing the layers and their activation functions. Convolutional Neural Networks were also tested out purely for the sake of seeing what those would output. CNNs are known for wide usage in Computer Vision, but quite a few articles studied in preparation for this project did mentioned the potentials of CNNs for text classification. However, upon various tests for the given problem, they did not seem to produce any fruitful results. The network structure described in the approach appeared to be the most optimal and best performing one out of all tested.

In addition to performing experiments of the model structure, the resulting model was also tested on a completely new dataset. Although, as always, there is a vast variety of song lyrics available, for the sake of this project we chose to explore the evolution of Taylor Swift [3] and how the genre of her songs has changed over the years. It is well-known that Taylor Swift initially became famous as a talented country singer. However, over the years her music shifted more towards pop, which is why she is a perfect test case for a model like ours. It is extremely difficult to detect a change in the genre just by exploring the lyrics. However, given our 65% accuracy we were curious to see what the model would output. As you can see in the Figure 3 below (it is interactive inside the jupyter notebook attached to this project), the points represent her songs sorted from the oldest (on the left) to the newest (on the right) up until 2017. It is clear that a lot of her earlier songs were, in fact, classified as either country or rock, and a lot of her latest songs were classified as pop or hip-hop/rap, which is a very interesting observation, and it once again proves the legitimacy of the model trained.

VII. Results

The goal of this project was to create an NLP model which could accurately evaluate a song's genre based only on its lyrics. After data preprocessing, we had a dataset with four labels. These labels were 'pop', 'rock', 'country', and 'hiphoprap'. In Figure 1 shows the accuracy of the model across each epoch. As you can see, the validation accuracy reaches 65% after seven epochs. The random chance accuracy should be 25% because there are four labels, and each label has the same number of songs associated with it. Our validation accuracy is much greater than the random chance accuracy.

Figure 2 shows the loss over each epoch. As you can see, the loss decreased greatly over time for our training set whereas the loss decreased until the third epoch for our validation set. Figure 2 suggests that we may have experienced some overfitting. The training loss greatly decreased each epoch, and after seven epochs, the validation loss seems to be relatively the same as it started. A possible reason for this is that we are training our model on similar data in each epoch. So, our model would get very good at recognizing the training data, but after a certain point, it may lose its ability to generalize. Another possible explanation is that, in general, this is a difficult classification problem. For most LSTM models, the training loss should always decrease across epochs. If we have a difficult classification, then we cannot expect the validation loss or validation accuracy to decrease much.

Initially, we were only seeing 20% validation accuracy and were using eight labels for classification. In order to improve the accuracy of our model, the first step we employed was to reduce the label set size to four. After reducing the number of labels in the data, we decided to employ a pre-trained model to encode our lyrics. The reduction of set size and use of a pre-trained model gave us our current accuracy of 65%.

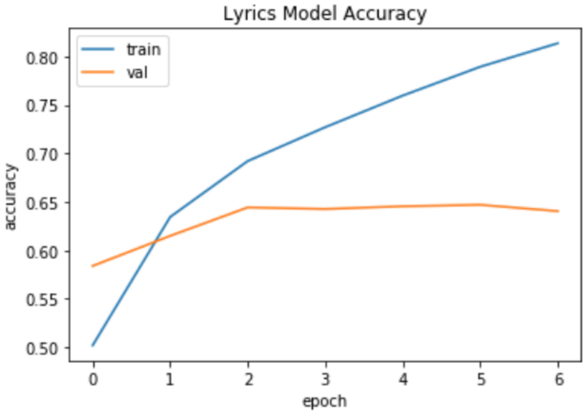

Fig. 1: Accuracy

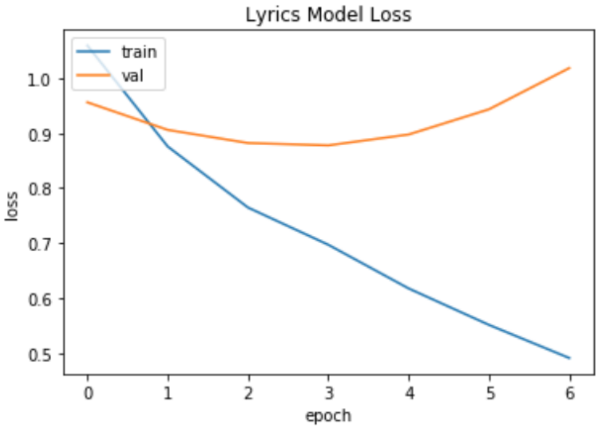

Fig. 2: Loss

VIII. Discussion

A lot has been done throughout this project, as it can be witnessed both throughout this report and the jupyter notebook itself. Although the final achieved accuracy is far from perfect, given the difficulty and complexity of the task, we can say that success was achieved throughout this project. Although further changes to the model structure should (and will) be made to improve the performance, 65% accuracy for the given problem is still quite a great achievement. The experiment of running the trained model on Taylor Swift model was also quite successful and helpful in understanding how the model works. Although the dataset did not exactly contain any ground truth labels, given our rich knowledge of Taylor Swift and her artistry, we were able to show how a model like ours could be used in practical situations.

IX. Conclusion

Genre classification is a difficult problem because we often rely on much more than lyrics to correctly classify the genre like melody, artist, and producer. However, our model suggests that there is an existing pattern in the lyrics we analyzed which can help us correctly identify their genre. This result is intriguing for companies that rely on genre classification for recommendations. It would simplify the feature space by which we classify different songs and allow for potentially accurate recommendations based on genres and subgenres.

In the future it would be interesting to see if this reduction in feature space gives a significant speedup in the classification problem. This could show just how useful our classification technique is. It would also be interesting to include just a small amount of metadata (like Artist name) in an attempt to increase the accuracy of the classifier.

Additionally, if we would like to expand this model to be used for song recommendation, we will need to add several more genres. We would like to add several genres to our classification without significantly decreasing the validation accuracy of the classifier. Because we saw overfitting in our model, we think that using a bagging algorithm could help to decrease the variance while keeping the bias relatively the same. Taking advantage of a confidence metric could also help with a recommendation service. If we had the model's confidence in each classification, we could set a threshold and only recommend songs that are above that threshold. This would allow us to be confident that the songs we are recommending are actually quite similar to the songs that our listener is already listening to.

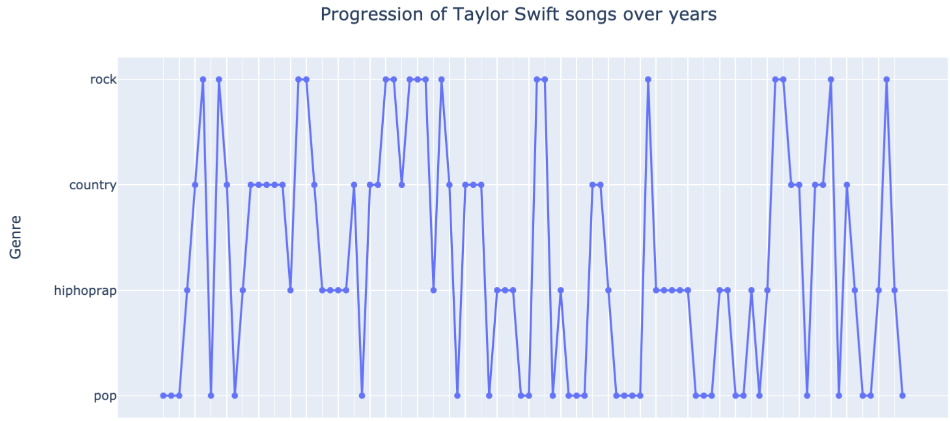

Fig. 3: Taylor Swift Experiment results

References

[1] M. Fell and C. Sporleder, "Lyrics-based analysis and classification of music," in *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*. Dublin, Ireland: Dublin City University and Association for Computational Linguistics, Aug. 2014, pp. 620–631. [Online]. Available: https://www.aclweb.org/anthology/C14-1059

[2] Kaggle. Lyrics dataset. [Online]. Available: https://www.kaggle.com/detkov/lyrics-dataset#songs_dataset.csv

[3] ——. Taylor swift dataset. [Online]. Available: https://www.kaggle.com/PromptCloudHQ/taylor-swift-song-lyrics-from-all-the-albums

[4] J. P. G. Mahedero, A. MartÍnez, P. Cano, M. Koppenberger, and F. Gouyon, "Natural language processing of lyrics," in *Proceedings of the 13th Annual ACM International Conference on Multimedia*, ser. MULTIMEDIA '05. New York, NY, USA: ACM, 2005, pp. 475–478. [Online]. Available: http://doi.acm.org/10.1145/1101149.1101255

[5] A. Tsaptsinos, "Lyrics-based music genre classification using a hierarchical attention network," *CoRR*, vol. abs/1707.04678, 2017.