Milestone 1

ABHINAV GHAI and ALEX TANASESCU

ACM Reference Format:
Abhinav Ghai and Alex Tanasescu. 2023. Milestone 1. 1, 1 (March 2023), 3 pages.

1 INTRODUCTION

This project will be an exploration into analysis of music lyrics. The primary goal will be to apply topic modelling techniques on lyrics to identify unique words or phrases that are most identifiable with a specific theme. Secondly the project will look into producing song embeddings using doc2vec and comparing similarities between songs within a genre, songs between genres, and songs between artists, using cosine similarity. Finally, we plan to represent any similarities or difference between the song lyrics, genres and artists using t-distributed stochastic neighbor embedding (t-SNE)

2 SOLUTION

2.1 Solution for the first part

For the first part of the project that focuses on identifying words and phrases that are identifiable with specific genre, we utilized TF-IDF and LDA. TF-IDF or term frequency-inverse document frequency is a relatively simple statistical technique that can be used to produce words or sets of words that correspond to a corpus. This technique makes use of the frequency of unique words and penalizes words that appear too frequently across the corpus, thus words that carry more 'unique' and 'relevant' meanings can be identified using TF-IDF. The corpus being used here are the lyrics of the different songs for a specific artist, therefore applying TF-IDF to it aims to theoretically identify the most significant words for these artists, which we can later use to assign topics [1]. We also utilize LDA, or Latent Dirichlet allocation to further produce another similar set of words for topics, which can help affirm our results from TF-IDF. LDA is a statistical model that effectively creates groups of groups. In particular when applied to a corpus which contains lyrics from a specific artist, this allows us to create a group of words that correlate to the groups of topics and themes in the artist's lyrics. LDA models contain a parameter called alpha that can be adjusted based on a priori about the distribution of topics, however given that we don't such a priori, we leave the parameter to be 'auto', which effectively creates this priori based on the corpus. In theory this should allow the LDA to effectively identify these topics and their corresponding words, we also chose the chunk size to be the default 2000, as no artists we analyzed had more than 2000 songs, so it fairly considered the entire corpus for every artist. For both these techniques preprocessing the data was an integral part of producing relevant and useful results.

Authors' address: Abhinav Ghai; Alex Tanasescu.

Manuscript submitted to ACM

2.2 About the datasets

We initially started off by considering three datasets:

- https://www.kaggle.com/datasets/deepshah16/song-lyrics-dataset
- https://www.kaggle.com/datasets/nikhilnayak123/5-million-song-lyrics-dataset
- https://www.kaggle.com/datasets/neisse/scrapped-lyrics-from-6-genres?resource=download

All three of these datasets posed a challenge that we hadn't anticipated in that they contained lyrics corresponding to non-English languages, however there was no easy way to identify if a specific song corresponded to English. Furthermore, our largest dataset was roughly 10 gigabytes in size, which made it difficult to work with when it came to experimenting with pre-processing and analysis techniques, but also contained mostly rap and pop songs, which made it so indiscriminately utilizing the entire dataset would likely lead to biases. Our 3rd dataset, wasn't nearly as large, however had the issue of having a large number of non-English songs as well, which made it hard to utilize. Our first dataset, was roughly 200 megabytes in size, and also contained artists which had mostly English lyrics. to the point where cases of the lyrics not being English didn't massively affect the results, as almost every artist had at least a 100 song lyrics, and for every case fewer than 1% of the songs were not in English. This dataset contained information about 21 artists, however we ignored BTS, and most of the artists were from the pop genres. This resulted in some challenges which are discussed later.

2.3 Pre-processing the data

Given the problems mentioned earlier, we focused on cleaning the first dataset for now. Firstly, we used nltk to tokenize each of the lyrics, we then lemmatized these tokens, to try to grab the essence of each word. We removed the default English stopwords nltk provides, but in addition we removed the top 2.5% words present for every artist, in an effort to remove some of the common 'sound words' used in music . That percentage was chosen simply because it seemed to produce the most consistently relevant results across all artists. Larger percentages seemed to remove quite a few relevant words that the artists talked about, and smaller percentages left in words like "yeah" or "I" which carried limited meaning when it came to identifying themes. All these preprocessing techniques applied to the LDA, the stopwords were also utilized in the TF-IDF, and generally speaking it resulted in better results across the artists.

2.4 Brief Discussion of Part 2 and its visualization

We still plan on continuing to solve part 2 using gensim's doc2vec and conduct similarity comparisons using cosine similarity. In theory using doc2vec, we should be able to produce embeddings for songs from different artists. This should allow us to average the song embeddings for specific artists and produce an 'artist-embedding' and similarly by averaging the song embeddings in a specific genre we should be able to produce a 'genre-embedding', both of which can be further analyzed using cosine similarity to answer questions regarding how similar artist's works in a specific genre are as well as how close certain songs are to certain genres. Furthermore we plan on using t-SNE, which is a visualization technique that allows you to effectively reduce high-dimensional data into a 2-D space, which should theoretically allow us present our data in a more intuitive format.

3 CHALLENGES OR DISCUSSION OF CURRENT RESULTS

The first obstacle we encountered came from inexperience with understanding how to deal with an extremely large dataset. Furthermore, this dataset also contained a large portion of non-English songs. This posed a challenge as not

only would the large dataset make it extremely hard to experiment with preprocessing techniques and other analysis techniques due to the large computation cost, but we are also evaluating if the results were acceptable manually, which also posed another challenge. To address this we decided to avoid using this dataset, and also stick to English songs, as we are only equipped to evaluate whether the results are relevant for them. Unfortunately, our other dataset only covers about 20 artists, which fit in at most 3 genres, so we have decided to look into other datasets, but are most likely to write a scraper to find artists from a specific genres. On a similar note, as we work towards analyzing a larger dataset, it will become less feasible to manually check the relevance of our results, and therefore we will have to rely on a the conclusions from a smaller sample size to reflect the validity of our results. The second major challenge we encountered was 'sound words' which were parts of songs like "ooh" or "ohhoo" or "oohhooo", and were challenging to deal with as they don't lemmatize well, and they tend to be off by a few characters, which makes it impossible to remove them using frequency alone. Fortunately this was a relatively isolated issue, in that it only affected some artists, but these 'sound words' frequently showed up in the tf-idf and LDA results. To address this problem, we simply ignored these words, but we may later readdress this issue. However, overall our results from tf-idf and LDA were fairly consistent with the artists, and there was a tendency for both results to focus in on specific word that were easily identifiable with themes present in an artist's work.

REFERENCES

[1] Ian Freed. 2018. *Using Machine Learning to Analyze Taylor Swift's Lyrics.* Retrieved January 29, 2023 from https://www.codecademy.com/resources/blog/taylor-swift-lyrics-machine-learning/