LyRec - 6.864 Final Project Report

Gokul Kolady, Nicholas Ramirez, Saumya Rawat

Abstract

Major music streaming platforms such as Spotify, Apple Music, and SoundCloud utilize intricate song recommendation systems to suggest new music to users. Typically, these recommendation systems rely on information such as song genre, tempo, and similar users' song histories. This project utilizes song embeddings and sentiment analysis to create recommendations based on lyrical content. We utilized two different methods to produce song embeddings. One involved creating term frequency-inverse document frequency representations for the lyrical content in a song, feeding that representation into a neural network to predict a valence (a measure of positive/negative sentiment), and appending that valence score to the initial representation to create a complete song embedding. The second method used the same pipeline, but instead of using tf-idf, the initial lyrical content representations were comprised of mean GloVe-sourced word embeddings for the words appearing in a song's lyrics.

Each of these models were then used to produce song embeddings for several sets of songs centered around specific moods, artists, or genres. The potential of these embeddings for song recommendation were then assessed by comparing the similarities of embeddings for songs within specific recommendation-relevant categories through clustering and mean cosine-distance comparisons. Overall, we found that while neither method of lyrical content representation was optimal for valence prediction, both methods produced final embeddings that encoded for several dimensions of song similarity, with the GloVe-based method producing better results in terms of valence prediction and embedding-based recommendation potential. The achieved results demonstrate the value that lies in considering lyrical content as an additional dimension for song recommendation. Furthermore, the dataset generated for the purposes of this project may prove useful for other lyric-related NLP research that requires data centered around less popular artists.

1 Introduction

Modern song recommendation systems typically focus on user data such as playlist similarity when generating song recommendations. This approach lacks a consideration of some of the most relatable parts of music. Specifically, it doesn't account for the lyrics of a song which conveys both emotion and mood. Within music, the core expression for an artist is through writing lyrics. This is one of the main parts of the song we develop a personal relation to. As a result, having a song recommendation system that utilizes lyrical content will create a highly personal approach to music recommendation. This recommendation system will enable us to identify similar music that surprises the user since lyrical similarity doesn't exclusively account for the sonic qualities of a song. Furthermore, if a successful lyric-based recommendation system is created, it could potentially be incorporated into current song-recommendation systems that streaming platforms utilize in order to enhance recommendation quality.

2 Related Work

Various music listening platforms, such as Spotify and Apple Music, have a song recommendation feature. This feature is often complex and built using multiple users' curated playlists to determine what other users' playlists should also incorporate. In this work, we focus on recommending what one user should add to their current playlist given the lyrical content of songs in their playlist and a known plethora of songs and their lyrics. Our model focuses on lyrical content and sentiment; thus, we will further discuss previous work focused

on lyrics-based song recommendation with word embeddings and sentiment analysis of song lyrics.

Reevesman [1] discusses embedding song lyrics to measure the spatial relationship between various songs and subsequently recommend songs that are in close spatial distance to a given song, where spatial distance is given by the similarity of words used in the songs. He uses the musiXmatch dataset of 237,662 songs which appear in bag-of-words format resulting in some information loss. To build high dimensional embeddings for songs, he uses a Distributed Bag of Words Paragraph Vector (PV-DBOW) model and uses t-Distributed Stochastic Neighbor Embedding to plot relationships in a lower dimension. Reevesman's works successfully shows music of particular artists grouped in similar low dimension spatial locations; however, he notes how spatial relationship is just the beginning towards building a larger recommendation engine.

Another way to determine song similarity based on its word content is through sentiment analysis of its lyrics. Certain words and phrases can hold a positive or negative meaning which we can link to moods and thus recommend songs of a similar mood. Zubair [2] preformed sentiment analysis of Billboard Top 100 songs from 1958 to 2019 and noticed that lyrics have become more negative over the years. Lyrics from 2019 are 4 times more negative than those from 1979 as determined through the polarity of the top keywords in a song's lyrics. Sentiment could potentially be linked to societal events, and Zubair recommends further investigation in this area. Lyrical sentiment analysis can help determine decades of music.

While many people may listen to songs for the sounds and rhythms, lyrics still act as an important factor in determining the mood, sentiment, and meaning of a song. Language and words hold feeling, allowing us to use lyrics to determine sentiment of a song and recommend a user similar songs. Veas [3] used Word Clouds, Statistics Table, Frequency Comparison Plot of Words, VADER Sentiment Analysis, and the Genius lyrics dataset to evaluate the sentiment of Metallica songs over their 4 decades from the 80s to 10s. In addition to evaluating overall sentiments of songs, Veas visually evaluated the most common words used in lyrics for particular decades symbolizing the change in content of songs.

Various NLP models and strategies have been used to evaluate how lyrics can be used to determine content and sentiment similarities. We use these prior works as a basis to create a song recommendation tool that focuses solely on using the lyrical content as compared to sounds and playlist similarities used by current complex recommendation systems. Language holds feeling and meaning which we believe listeners value when choosing songs to listen to.

3 Methods

3.1 Data Generation and Pre-Processing

For our study, we predict sentiment given a song's lyrics. In order to accomplish this goal, we will need a dataset of many song lyrics and a corresponding sentiment label. To build this dataset, we used the Genius API and Spotify API. In order to get a large, diverse set of song lyrics, we used the Genius API artistID attribute. We determined the range of the artistIDs to be 2961456 and randomly generated numbers to build a set of artists. For these artists, we found all the artists' songs with the songIDs attribute. We used these songIDs to obtain the name, artist, and lyrics of the song. We pre-processed lyrics to remove anything with brackets (generally words like "chorus", "intro", "prelude", etc), remove punctuation, remove new lines, and lowercase all text. In order to retrieve sentiment labels for each song, we will use the Spotify API's valence attribute. Valence is in the range of 0 to 1. Valence of happier and cheerful songs are closer to 1 while valence of angry and sad songs are closer to 0. We found the valence attribute by providing the Spotify API with the name and artist of the song. This process resulted in a diverse random set of 9,159 songs with lyrics and valences (sentiment).

3.2 Embedding Creation

3.2.1 Term Frequency Representations

Term frequency was one of the key components of the lyrical content of a song that we used in order to represent it as an embeddings. We wanted lyrical content to be one of the prime factors utilized in evaluating song similarity for the purpose of song recommendation. However, a simple measure of term counts within a song doesn't account for the fact that the most common words in the English language likely dominate lyrical content across the board. Furthermore, these terms (such as "the", "and", and "I") don't divulge much unique information about the feeling, sentiment, or energy

of a song. Thus, we decided to implement term frequency-inverse document frequency representations using each song's lyrical transcript in order to reduce the impact of these frequent yet uncharged words on the song representations.

3.2.2 Valence Prediction

Of course, aside from term frequency, there are several characteristics of a song that influence the way it is perceived and therefore enjoyed by the listener. Some of these characteristics are included within the data that the Spotify API provides for each of its songs. Of these characteristics, the one that we felt would be most relevant to consider in our song representations was "valence." This is a term created by Spotify that is meant to quantify the happiness or sadness of a song. In other words, it is analogous to sentiment. We realized that if we wanted our song embeddings to be purely based on lyrical content, we would need a way to generate valence from lyrics. We thus created a neural network that took the term frequency representation of a song as input and produced its valence score as output.

Once the network was trained, in order to create a full embedding for a song we did the following: First, we generated its term frequency representation (TFR). We then fed this TFR into our trained neural network in order to produce the song's valence score. Finally, we appended that valence score to the song's TFR vector in order to construct the song's full embedding.

3.3 GloVe Word Embedding Representations (Alternative to TFR)

As discussed further along in this paper, we achieved sub-optimal performance when using TFR vectors to predict valence for songs. We suspected that this may have been a result of the lack of sentimental and contextual information that TFRs contain (as they are mostly concerned with the frequencies of terms rather than the meanings of those terms). Thus, we tried to create alternative song lyric representations that captured more information about the meanings of the words present in a given song. In order to accomplish this, we utilized pre-trained and pre-constructed global word embedding vectors provided by GloVe. Specifically, we used the GloVe trained 300 dimensional vectors from Wikipedia 2014 and Gigaword 5 which has 6 billion tokens and 400K vocab. Instead of creating a tf-idf vector for a song, we simply averaged the GloVe word embeddings of the words present in the lyrics of that song (we averaged over the words that were present in the GloVe vector set) in order to create its lyric representation. We then proceeded with the aforementioned pipeline of predicting a valence score and appending it to the lyric representation vector to create a complete song embedding, except we replaced the TFRs with GloVe representations at every step.

3.4 Tools Used

We utilized Python to create our song embeddings, including the construction of term frequency-inverse document frequency representations and GloVe representations, along with the use of those representations to fill out song embeddings with a predicted valence score. We used PyTorch to train, develop, and evaluate our neural networks and utilized Google Colab to facilitate these processes. Our model development involved high amounts of computation due to the nature and quantity of features that we aimed to generate and combine, thus a platform such as Colab Pro was necessary to ensure efficient model development. We chose the Genius API to source our lyrical data because its extensiveness, the fact that it provides raw lyrical content instead of a bag of words for each song (many online lyric databases use the bag of words approach), and its familiarity as a platform. We chose the Spotify API for evaluation because Spotify has one of the largest and highest quality collections of mood and genre based playlists. Finally, we utilized GloVe in order to create word embedding based representations of song lyrics, as it provided pre-made and meaningful word embeddings. We chose to derive word embeddings from GloVe due to its credibility as a source for meaningful embeddings and our desire to use reliable embeddings that would have a high likelihood of containing contextual and sentimental information.

Our implementation code can be found here: https://github.com/gokulkolady/lyrec_nlp.

4 Experiments

In order to evaluate our embeddings, we grouped songs based on a plethora of categories and determined whether groups of songs within those categories shared similar embeddings. These categories included traits such as genre, artist, and mood. The goal was to use these song group similarities as metrics to indicate whether our embeddings could

potentially serve as valuable tools for recommendation, as if neighbors within the embedding space represent songs that were similar in traits like genre, mood, or artist, they would likely serve as effective recommendation tools. To identify groups of songs within these categories, we utilized Spotify's API to retrieve playlists created around specific moods (Happy Hits, Sad Hour), genres (Country Mix, Rap-Caviar), and artists(Frank Ocean, Xxxtentacion). We ran three main experiments to test our model.

4.1 Predicting Valence (Neural Networks)

The first experiment tested the accuracy of the models we created to predict valence scores. Specifically, we trained the model on numerous tf-idf and GloVe representations of songs, and tested their ability to predict song characteristic labels/values sourced from Spotify. For our neural network setup, we found the following hyperparameters as optimal for decreasing our testing loss. Our neural net contained two layers with a ReLU activation function, a mean squared error loss function, and Adam for an optimizer algorithm with a learning rate of 0.001. We trained our neural network for 50 epochs with a batch size of 32. In addition, we utilized 10-fold cross validation (the final loss values shown are the average of the cross validations).

TF-IDF to Valence Performance	
Training Error	4.3961
Test Error	24.5981
Baseline Error	20.2337
Better Guess Frequency	0.44993

Table 1: Performance of our model when using TF-IDF representations.

In Table 1, "Training Error" represents our model's mean absolute error (MAE) for the training set over the course of 10-fold cross validation, "Test Error" represents our model's MAE over the test set, "Baseline Error" represents the baseline model's MAE over the test (where the baseline model simply predicted the mean valence of the test set for every test sample), and "Better Guess Frequency" is the frequency with which our model had a smaller MAE than the baseline model on the test set. As can be seen, the Test Error was not ideal, as it indicates that our model performed worse than simple baseline model that predicted mean valence across the board. This is further confirmed by the fact that the Better Guess Frequency was lower than 50%. By comparing the Training and Test Errors in Table 1, one might suspect that our model was over-fitting on the training set. We had the same suspicion, and thus attempted to prevent over-fitting several methods such as dropout, changing the epoch count, changing the training batch size, and changing the number of network layers, and while these methods made the Training Error higher, they had a negligible effect on the Test Error.

We suspected that this poor performance was either a result of a lack of correlation between term frequency representations and Spotify's song valence scores, or a lack of correlation between lyrics as a whole and Spotify's valence scores. As a sanity check, we decided to test the correlation of the term frequencies of certain intuitively positive or negative-sentiment words with Spotify valence scores:

Figure 1: Correlation between words and valence for TF-IDF Representations.

Clearly, Figure 1 shows that term frequencies are not usefully correlated with Spotify valence scores. If they were, we would see terms like "happy" and "death" have a much higher magnitude of correlation with valence, as these should be terms that are extremely predictive of sentiment. If these terms are so unreliable in their predictive power, then more subtly sentiment-charged terms would likely be even less useful in neural network prediction. While this correlation matrix displays evidence against the usefulness of tf-idf representations in this context, it does not necessarily discount the potential correlation between lyrical content in general with valence. Thus, we decided to try predicting valence with word-embedding representations derived from GloVe instead of using term

frequencies.

Our neural network setup for GloVe representation was the same as the tf-idf representations mentioned above at the top of this section. Here are the performance metrics for predicting valence from GloVe representations.

GloVe to Valence Performance	
Training Error	18.0683
Test Error	18.7596
Baseline Error	20.2187
Better Guess Frequency	0.557518

Table 2: Performance of our model when using GloVe-based representations.

The metrics shown in the table 2 have the same meanings as those in Table 1. Upon comparison between Tables 1 and 2, it is evident that the GloVe-based representations proved more useful in this predictive task. Unlike the tf-idf model, the GloVe-based model was able to produce test results better than the baseline model on both metrics. The Test Error was lower than the Baseline Error and the Better Guess Frequency was above 50% (notably, it was over 10% better than that of the tf-idf model).

4.2 Song Embedding Scatter Plots

The second experiment created scatter plots for song embeddings created from both TFRs and GloVe. These song embeddings were created from our training corpus (9159 songs) as well as our evaluation songs. The creation of our training corpus is described further in section 3.1. After generating embeddings for all of the songs in our training corpus, we utilized T-distributed Stochastic Neighbor Embedding (TSNE) to visualize and analyze the embeddings. We attempted to utilize principal component analysis (PCA) for this visualization, however the nature of TFRs led to non-optimal visuals. Specifically, for a given song embedding (created with TFRs) with 2054 dimensions, there were on average 2007 dimensions with zero values. Thus, the loss of information from PCA reduction was too significant to visualize the song embeddings well. For the first scatter plot visual, we plotted the TFR song embeddings for the top 8 artists (artists with the most songs in the dataset) from the entire training corpus.

As shown in Figure 2, we can see a wide range spread of songs from similar artists. On the left side, we can see three separate clusters around the

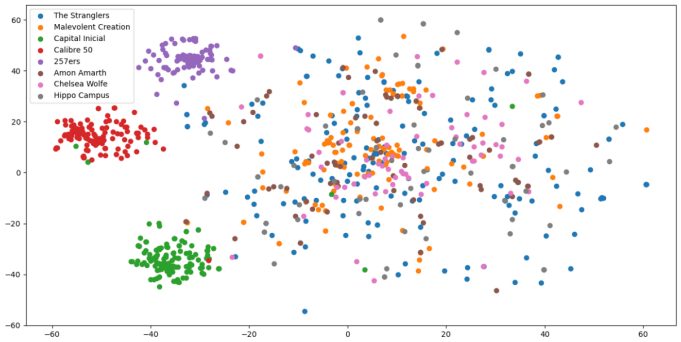

Figure 2: TSNE Image of Top 8 artists' song embeddings utilizing TF-IDF.

same artists. Upon researching a few songs in each cluster, we determined that the clusters are mostly formed based on language. However, we can see some outliers within this where different artists are spread across the clusters for a couple of songs. The main sparse cluster in the center represents English artists, and there doesn't appear to be a specific pattern for the artists within this cluster. We also created a scatter plot of our song embeddings created by GloVe to see the difference.

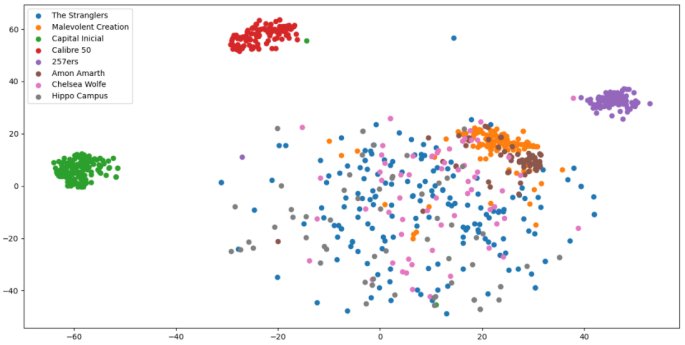

Figure 3: TSNE Image of Top 8 artists' song embeddings utilizing GloVe.

As shown in Figure 3, there are more clear distinct visual clusters formed than the previous visualization with TFRs. These clusters are also mostly separated by language, and have fewer outliers within each cluster. The larger cluster in the center also represents mostly English songs. We also see more better patterns within this English cluster as Orange (Malevolent Creation) is much denser. The smaller clusters around it represent languages such as Spanish, German, and Portuguese. Generally, people listen to songs of the same language so these song embeddings would provide a way to recommend songs of the same language.

Both scatter plots reveal that language is a major factor within song embeddings. However, it is

clear that GloVe embeddings provide a significantly better representation of lyrical content if the goal is song recommendation. This better representation is shown with tighter and denser clusters around specific artists as well as more distinct clusters in general. In addition to the training corpus, we created scatter plots to show how song embeddings differentiate between genre, artist, and mood. The first shown is genre based, specifically, we created song embeddings from the Spotify playlist Country Mix and RapCaviar. We will only look at GloVe song embeddings as they provide the most insight.

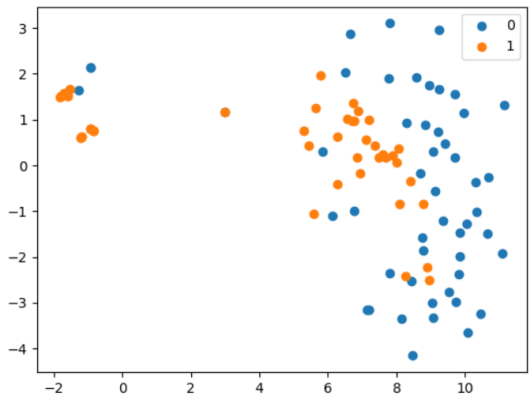

Figure 4: TSNE Image of Genre song embeddings utilizing GloVe and predicted valence.

In Figure 4, the orange (labeled as 1) represents songs from the Country Mix playlist and the blue (labeled as 0) represents songs from the RapCaviar playlist. The scatter plot reveals a small separation between the two genre-based playlists. However, it's clear that there is no solid separation between the two genres. As a result, we can draw that utilizing song embeddings from GloVe wouldn't create great recommendations for similar sounds. However, the lyrical content would be similar.

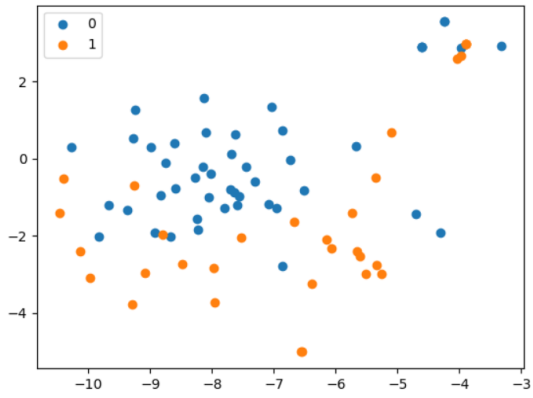

Figure 5: TSNE Image of Artist song embeddings utilizing GloVe and predicted valence.

In Figure 5, the orange (labeled as 1) represents songs from the This is Frank Ocean playlist and the blue (labeled as 0) represents songs from the This is Xxxtentacion playlist. We see that the orange is generally towards the bottom and the blue is towards the top indicating a slight separation between the two artists. GloVe was able to slightly tell the difference between the two artists; however, it isn't very distinct showcasing that it's difficult to differentiate between artists, who have different sentiment levels, given the lyrical content.

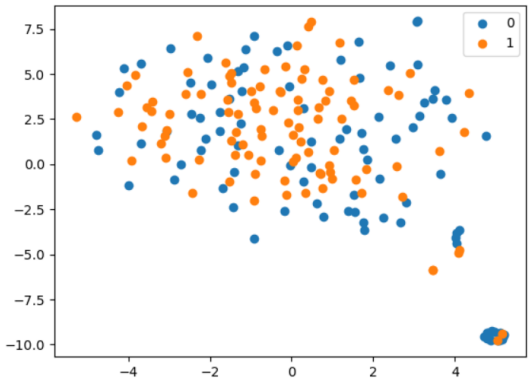

Figure 6: TSNE Image of Mood song embeddings utilizing GloVe and predicted valence.

In Figure 6, the orange (labeled as 1) represents songs from the Happy Hits playlist and the blue (labeled as 0) represents songs from the Sad Hour playlist. This playlist is much more mixed and has no clear patterns within the song embeddings. This is most likely caused by our predicted valence values due to the little correlation between key words and valence. In addition, the playlist is split by mood so they mostly share a very similar overlap of genre, artists, and overall style.

4.3 Playlist Embedding Similarities

The last experiment involved calculating average pair-wise distance between the embeddings of songs in different Spotify playlists to evaluate embedding similarity between categorically similar songs. Essentially, we evaluated how similar the embeddings for songs were within the same mood, artist, or genre-based playlist by evaluating the average pair-wise euclidean distance within the playlist. We then compared it to the average pair-wise euclidean distance within a test set that contained all of our test playlist songs. Finally, we generated the mean pair-wise distance between songs in our valence prediction training set. This helped us evaluate how well we were able to extract important

characteristics of a song from lyrics, and allowed us to assess which aspects of song similarity our embeddings effectively account for.

First, we ran this distance evaluation on our embeddings generated by appending valence scores (predicted using the tf-idf neural network) to tf-idf representations. As the valence scores had been scaled to 0-100 for network training/prediction, we centered them around 0 and scaled them back down in the final embeddings by subtracting 50 from the valence scores and dividing them by 10,000 before appending them to the tf-idf vectors. The playlists we used comprised 3 categorical pairs: mood (happy and sad), genre (country and rap), and artist (Frank Ocean and XXXTentacion).

Song Grouping	Euclidean Distance
Happy Hits	176.3528
Sad Hour	106.2270
Both Happy and Sad	144.8477
Country Mix	154.8716
RapCaviar	296.8880
Both Country and Rap	232.4831
This is Frank Ocean	144.6663
This is XXXTentacion	167.7756
Both Frank and X	154.9655
All Playlists Combined	171.0478
Valence Training Corpus	102.7278

Table 3: Mean pair-wise euclidean distance scores for TFR-and-valence embeddings within different song groupings.

As can be in Table 3, the results were mediocre at best. Ideally, the average pair-wise euclidean distances for each individual playlist would be lower (and thus more similar in terms of song embeddings) than those of their respective categorical pairs, All Playlists, and the Valence Training Corpus. This would be desirable as it would indicate that songs grouped within specific categories have more similar embeddings. For each categorical playlist pair, only one member of the pair was more similar than its respective pairing, which is not optimal. 4 out of 6 individual playlists were more similar than All Playlists Combined, which is more optimal. Most notably, none of the individual playlists were more similar than the Valence Training Corpus (which including over 9,000 mostly randomly-selected songs), which is very sub-optimal.

Next, we ran the same distance evaluations on the same song groupings for our embeddings generated by appending valence scores (predicted using the GloVe neural network) to GloVe representations. We performed the same adjustment on valence scores for these embeddings as we did for the TFR-and-valence embeddings.

Song Grouping	Euclidean Distance
Happy Hits	2.1061
Sad Hour	1.7042
Both Happy and Sad	1.9461
Country Mix	1.9002
RapCaviar	1.8630
Both Country and Rap	1.9931
This is Frank Ocean	1.8317
This is XXXTentacion	2.4160
Both Frank and X	2.1191
All Playlists Combined	2.0360
Valence Training Corpus	2.9039

Table 4: Mean pair-wise euclidean distance scores for GloVe-and-valence embeddings within different song groupings.

These results are significantly more desirable than those produced by the TFR-and-valence embeddings. For the categorical playlist pairs, one member of the pair was more similar than its respective pairing for 2 pairs, and both members of the pair were more similar than their respective pairing for 1 pair (compared to 3 and 0 respectively for TFR-and-valence representations). 4 out of 6 individual playlists were more similar than All Playlists Combined (same as the TFR-and-valence representations). Most importantly, *all* of the individual playlists were more similar than the Valence Training Corpus (compared to none for the TFR-and-valence representations).

5 Conclusion

When comparing the results generated by tf-idf representations and GloVe representations for valence prediction, scatter plotting, and euclidean embedding similarities, it seems that GloVe-based representations perform better in most aspects. GloVe-based lyric representations performed better on both test-accuracy metrics in predicting valence, created more clear valence-appended song embedding clusters among the training corpus, and resulted in embeddings that encoded for song characteristic similarity better according to the euclidean distance metrics. This was the outcome that we

expected, as while tf-idf representations solely account for term frequencies across song lyric documents, our GloVe representations contain information about the contextual and sentimental meanings of the words present in those documents, and is still able to capture the relative frequencies of these terms within lyric documents through weighted averaging of GloVe vectors.

Even using GloVe based representations, valence prediction performance was not optimal. Although it was slightly better than the baseline model performance, there is still room for improvement in terms of mean test error. This is most likely a result of the fact that Spotify's valence scores for songs are not well-correlated with their lyrical content. That being said, the rest of our Glove-based embeddings (excluding valence) seemed to encode for meaningful song representations that captured song characteristics such as mood, genre, and artist. The characteristic it seemed they seemed to best capture was genre (the individual genre playlists were more self-similar than the pair of them combined), which is ideal, as genre is one of the most important factors in recommendation quality. All in all, our research demonstrates that there is significant potential in utilizing song lyrics as a dimension for song recommendation, and we believe that major streaming services should look into incorporating this dimension into their recommendation systems.

6 Future Work

There are a few areas of improvement we would like to address in the future. The Spotify API offers a variety of song characteristics (valence, danceability, genre, etc). We will explore which combination of these song characteristics results in the best representation of songs. Additionally, we can apply our model using different datasets. We will explore using a dataset that only includes only popular songs (songs may be better labeled) and a dataset focused on including a similar distribution of songs based on language.

Our model will be used to recommend songs. A user would be able to input a song, we would find the lyrics of the song and create an embedding, and use nearest neighbors to determine what songs are similar and thus should be recommended to the user. We will compare our lyric-based recommendation to those of other recommendation systems which typically include more information such as sounds, genre, artist, and tracked user data. If we can provide a recommendation system based solely on lyrics, then we can limit the amount of data we track from user's music listening behavior and provide an added layer of privacy.

References

[1] Reevesman, Adam. "Lyric-Based Song Recommendation with doc2vec Embeddings and Spotify's API." Medium, Towards Data Science, 19 July 2020, https://towardsdatascience.com/lyric-based-song-recommendation-with-doc2vec-embeddings-and-spotifys-api-5a61c39f1ce2.

[2] Zubair, Salim. "Sentiment Analysis of All Billboard Hot 100 Songs over Time (1958–2019)." Medium, Towards Data Science, 3 Jan. 2020, https://towardsdatascience.com/sentiment-analysis-of-all-billboard-hot-100-songs-over-time-1958-2019-3329439e7c1a.

[3] Veas, Cristóbal. "How to Analyze Emotions and Words of the Lyrics from Your Favorite Music Artist." Medium, Towards Data Science, 25 Sept. 2020, https://towardsdatascience.com/how-to-analyze-emotions-and-words-of-the-lyrics-from-your-favorite-music-artist-bbca10411283.