A Statistical Analysis of Song Lyrics in Personal Spotify Libraries

Raymundo Mora
School of Data Science
University of Virginia
Charlottesville, USA
rm3xw@virginia.edu

***Abstract*—The repository for this project is hosted here. In this project we explore how we can use natural language processing tools to better understand the texts we consume as media. Particularly, we look at the author's personal *Spotify* library in this project but the repository hosting this project can easily be cloned to allow any user to repeat the analysis with their own data. In this project we find patterns in the lyrics of the songs that the author enjoyed enough to add to their library in order to gain insight about listening patterns and relationships between artists in the corpus.**

***Index Terms*—Natural Languge Processing, Text Analytics, Pincipal Component Analysis, Topic Modeling, Hierarchical Cluster Analysis**

I. Introduction

The lyrics of a song can tell an emotional story in the matter of minutes. It can make someone go through a roller coaster of emotions that cause physiological responses like a change in heart rate or perspiration. The lyrics of a song can illicit old memories or create new ones through new associations. The words of our favorite songs can say a lot about us, our experiences, and how we individually view music. This is part of the motivation behind the team's desire to explore the lyrics of the music stored in their *Spotify* library. The notebook *1. Data Collection.ipynb* is developed so that anyone can do the same with their own library using the code that compliments this report.

II. Data

A. Collection

All the data for this project are hosted here. You may download the data which contains the songs downloaded the author's *Spotify* library before April 5, 2023. The files should all be put in the *datasets* folder located at the root of the project repository.

The analysis we ran, and all of the notebooks generated can be ran by downloading songs and their information using the *Spotify API*. The process can be replicated by downloading a list of predefined tracks or by searching for N tracks from an artist, album, or public playlist. Since the we wanted to take a look at all of the songs they have liked in their personal library we had to go through some authentication steps. These steps are all highlighted and walked through in the 'Getting Started' section of the README.md file for this project. At the time of this analysis *Spotify* did not provide a method for downloading lyrics through their API. To get around this we used the 'Genius API' to get lyrics for the songs obtained from the *Spotify API*. The steps for authenticating yourself for the *Genius API* are also found in this repository's *README.md*.

B. Data Overview

Using the *Spotify API* we download all of the songs in our personal library. The songs are fetched 50 at a time since this is the limit for the API. Each song has a unique *track.id* and other metadata such as the track number in its corresponding album and the song popularity. Each song and all of its metadata is put into *LIB.csv*, where rows are individual songs and the columns are features of the metadata.

Next, we generate our *F1_CORPUS.csv* file by iterating through every song in *LIB.csv* and use the song and artist name to obtain the HTML containing lyrics for the song through the *Genius API*.

We are still missing some useful information about the songs in our library so we use the *Spotify API* once again to add a list of associated genres as a feature of each song. It is important to note here that there is no direct way to get genres for songs through the API so we use the genres associated with the principal artist of each song to add its genre data.

A high level exploration of our data reveals that there are 12 languages present in our corpus with the most prominent being English followed by Spanish (see Figure 1). Figures 3 and 2 reveal the amount of occurrences of the 20 most common genres and artists present in our corpus.

We obtain language information by using the python package *langdetect* [Danilak(2021)]. This python package allows us to map each track to the language it was written in. We find that of the 1910 songs in the library 1447 (76%) of them are in English and 442 (23%) of them are in Spanish as seen in Figure 1. In order to avoid the complications that may come with comparing texts across multiple languages, we move forward by only considering the tracks whose lyrics are written in English.

We proceed to augment our data by tokenizing it and defining our Ordered Hierarchy of Content Objects (OHCO) with the following sequence: **song, section, line,term**. Using this OHCO, we define sections as individual blocks that can contain verses, bridges, and choruses of a song. Lines are defined as individual lines of lyrics as written by *Genius*. To

avoid the tokenization of punctuations or special characters like emojis/emoticons, we strip every line in our corpus of non-alphanumeric characters.

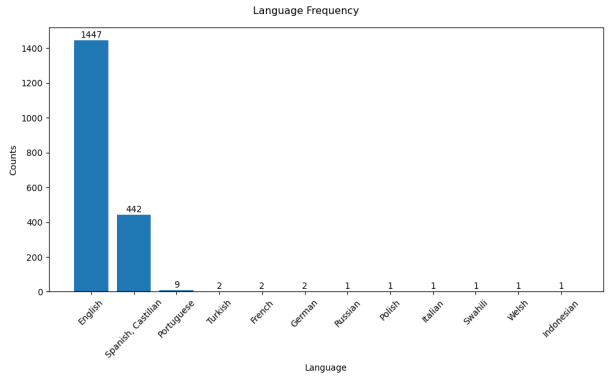

Fig. 1. Tracks by Language

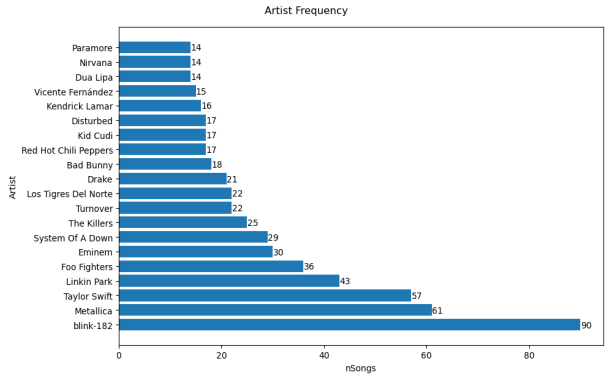

Fig. 2. Frequency of 20 Most Repeating Artists

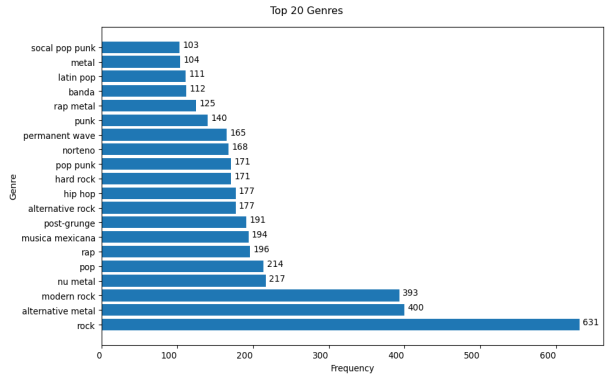

Fig. 3. Frequency of 20 Most Repeating Genres

III. Text Analysis

A. Principal Component Analysis

We use principal component analysis (PCA) to reduce the dimensionality of our data and look for clusters that could help define clear distinctions in lyrical content between artists. For this part of our analysis we use songs as our bag of words. Songs are chosen because due to the nature of our corpus, choosing levels futher own our OHCO will generate small bags without much information. Choosing songs as our bags allows for themes to emerge from the story each song is trying to tell and map out where the songs of each artist lie inside a vectorized space. Our corpus contains close to 500 unique artists with songs written in English. Not all of these artists contain a large sample size within our corpus and visualizing distinct clusters between them in our PCA could get clustered.

For the reasons just listed, we perform our PCA by limiting our data to only the songs from the 10 most frequently appearing artists in our corpus. Making this decision makes our result visualizations more interpretable and excludes collections of texts that do not have significant sample sizes. In Figure 4 we plot the first two principal components. Figure 4 reveals that artists like Drake, Taylor Swift, and Metallica have the largest ranges in the first principal component.

Through this analysis we also see that Eminem and Drake are share similar space in these dimensions. Both of these artists are some of the most successful rappers on the planet and although they have different styles it is interesting to see them share this space lyrically.

The Killers and Turnover are the two groups who have the tightest clusters in both dimensions. These two artists have distinct sounds but through PCA we are able to see how they share this space and tend to not vary much lyrically.

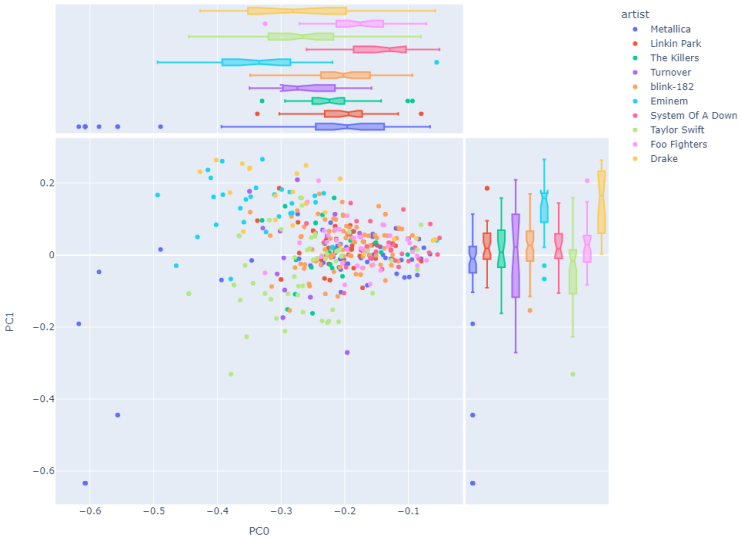

Fig. 4. PC0 vs PC1 Top 10 Artists

In Figure 5 we only plot the principal components of our top 3 artists by song occurrences in the corpus. This plot reveals three distinct clusters for our artists with Taylor Swift and Blink-182 clearly having their own spaces. Not only are the styles between these to artists vastly different but the content of their lyrics generally encompass two very different genres

so it is good to see this clear distinction. Metallica is an artist that has been around for much longer than both of them which may be a reason that it has a much larger absolute range range in both dimensions. Although Metallica has a large absolute range, many of their songs look like they have a variance that can be explained by the first two principal components similarly to Blink-182. These two artists perform different sub-genres of rock. Further if we get the topic best associated with an artist (see Figure **??**)

It is important to note that when we are talking about these spaces and the range of artists in the analysis we are only considering songs contained in our personal library and not the entirety of their discography.

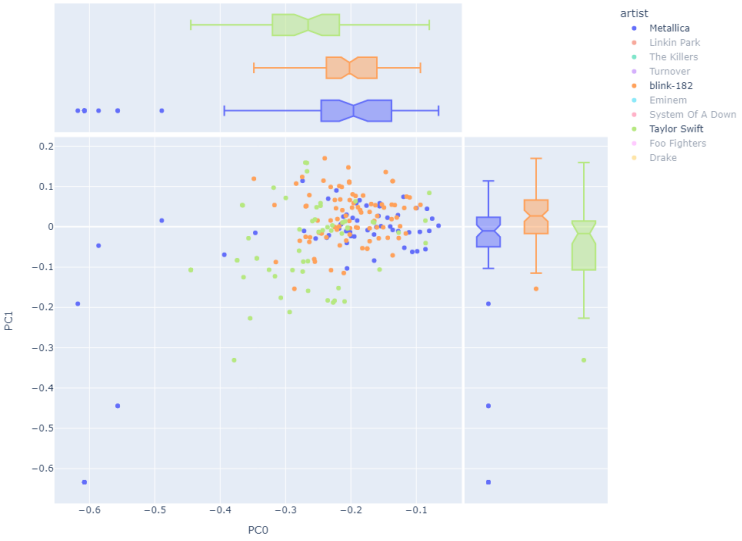

Fig. 5. PC0 vs PC1 Top 3 Artists

B. Topic Models

We use Latent Dirichlet Allocation (LDA) modeling to generate topics for our corpus. We have labels for the genre of each song based on the artist, however, these labels are lists of all the genres that those artists fall into. Instead of generating topics for each genre in our corpus we group our data by artist so that we may generate a vector of each topic per artist. We have close to 500 artists and chose to generate 50 topics through our model and return the 10 most significant terms. Figure 6 shows a subset of what our topic model returns.

We test how useful and accurate our topic model is in Figures 7 and 8. We do so by looking at the topics that are best associated with the artist Turnover. Once we have the top associated topics we can take the label of a topic, for example **T37** as seen in the Figures, and look for the songs that are most associated with that topic.

In this context, deploying something like this for our local library could serve as a recommendation system. In this type of recommendation system one could give the model an artist that exists in the library and queue related songs that also exist in the library.

artist	blink-182	Metallica	Taylor Swift	Linkin Park	Foo Fighters	Eminem	System Of A Down	The Killers	Turnover	Drake
T00	0.008197	0.006242	0.000834	0.005128	0.000882	0.005655	0.026679	0.002966	0.015557	0.003306
T01	0.012399	0.019782	0.001465	0.017364	0.049307	0.017747	0.025711	0.028728	0.000719	0.000346
T02	0.007669	0.012876	0.019642	0.005996	0.023819	0.131312	0.017044	0.006845	0.009605	0.090099
T03	0.023569	0.028052	0.053389	0.004642	0.041218	0.014981	0.000629	0.039480	0.054569	0.065372
T04	0.004506	0.000455	0.031855	0.000683	0.037598	0.036153	0.053337	0.003041	0.000719	0.003598
T05	0.071523	0.015487	0.022007	0.010982	0.026332	0.006720	0.003861	0.006794	0.026446	0.020926
T06	0.006245	0.004171	0.007032	0.013295	0.068023	0.000187	0.005162	0.055950	0.033217	0.000346
T07	0.007957	0.017249	0.018123	0.000683	0.007603	0.003660	0.004071	0.002405	0.019440	0.041216
T08	0.000802	0.004778	0.012906	0.005147	0.000882	0.032643	0.016762	0.000498	0.000719	0.134293
T09	0.063007	0.016726	0.040873	0.009153	0.012670	0.006390	0.008818	0.008603	0.084188	0.002012
T10	0.005753	0.018275	0.000808	0.034413	0.000882	0.012707	0.000629	0.001059	0.000719	0.000346
T11	0.010236	0.043497	0.004265	0.000683	0.000882	0.025922	0.044058	0.010002	0.007068	0.020507
T12	0.003729	0.013329	0.001442	0.026017	0.006121	0.019530	0.000629	0.006562	0.000719	0.185388
T13	0.002021	0.017117	0.013386	0.008836	0.000882	0.000187	0.000629	0.006191	0.000719	0.000346
T14	0.009279	0.003018	0.000916	0.001875	0.000882	0.022073	0.000629	0.004041	0.000719	0.000346
T15	0.004870	0.025235	0.029351	0.017534	0.003486	0.000187	0.000629	0.010164	0.000719	0.038763
T16	0.018761	0.000455	0.005036	0.059759	0.000882	0.000187	0.002601	0.000498	0.002937	0.000346
T17	0.023945	0.023677	0.010788	0.028652	0.028719	0.003628	0.034370	0.013042	0.002014	0.007147
T18	0.024555	0.031007	0.027302	0.162839	0.049952	0.001731	0.008026	0.039226	0.020413	0.000346
T19	0.009669	0.014385	0.012268	0.000683	0.017303	0.002212	0.000629	0.003827	0.000719	0.002260
T20	0.004799	0.097994	0.003303	0.000683	0.000882	0.009023	0.000629	0.000498	0.025935	0.000346
T21	0.073803	0.019766	0.068074	0.071474	0.018881	0.005420	0.115815	0.060235	0.060356	0.015459
T22	0.002630	0.013839	0.028109	0.048235	0.047060	0.003244	0.004457	0.002976	0.000719	0.001054
T23	0.002977	0.009423	0.011158	0.007151	0.069305	0.002238	0.033560	0.017508	0.010971	0.002394
T24	0.011453	0.021657	0.025778	0.000683	0.000882	0.135398	0.035351	0.006527	0.000719	0.044551
T25	0.021297	0.000455	0.007572	0.000683	0.005999	0.058644	0.013537	0.003523	0.000719	0.024299
T26	0.041513	0.045142	0.103077	0.081510	0.037453	0.001509	0.000629	0.008159	0.113715	0.016016
T27	0.012052	0.000455	0.012502	0.006637	0.001913	0.006646	0.053458	0.013249	0.010262	0.005748
T28	0.000602	0.002225	0.007666	0.000683	0.004417	0.000187	0.000629	0.000498	0.000719	0.000346
T29	0.000602	0.033229	0.006010	0.003395	0.004282	0.001306	0.000629	0.015130	0.007324	0.000346
T30	0.045563	0.030488	0.052365	0.007723	0.037880	0.057585	0.027608	0.177252	0.031597	0.033497
T31	0.009282	0.000455	0.000440	0.070731	0.041817	0.049470	0.000629	0.008308	0.000719	0.004676
T32	0.022271	0.034468	0.022883	0.037810	0.005721	0.001465	0.032422	0.004168	0.026924	0.002903
T33	0.008754	0.025399	0.002462	0.010833	0.000882	0.012884	0.028497	0.009552	0.000719	0.000346
T34	0.014825	0.018202	0.004830	0.015948	0.050877	0.027964	0.042554	0.039868	0.000719	0.001187
T35	0.004728	0.050648	0.000440	0.029187	0.023219	0.001500	0.039277	0.016844	0.000719	0.000346
T36	0.006710	0.003376	0.012080	0.002169	0.005828	0.003644	0.000629	0.014513	0.006698	0.012495
T37	0.021214	0.043233	0.034605	0.017039	0.044987	0.003480	0.023162	0.085203	0.140195	0.004432
T38	0.004865	0.002137	0.008043	0.001881	0.000882	0.005675	0.020945	0.000498	0.000719	0.004167
T39	0.009819	0.021296	0.034255	0.031905	0.027819	0.000986	0.014705	0.001237	0.010550	0.000346
T40	0.038536	0.036182	0.002056	0.000683	0.021246	0.003131	0.000629	0.033700	0.000719	0.008239
T41	0.130215	0.030796	0.155727	0.056286	0.068215	0.077503	0.000629	0.097091	0.069658	0.057590
T42	0.030955	0.030787	0.022823	0.002320	0.008561	0.073836	0.047568	0.032188	0.000719	0.121928
T43	0.013094	0.000455	0.010418	0.000683	0.000882	0.014567	0.000629	0.000498	0.000719	0.005352
T44	0.021383	0.007095	0.006158	0.017383	0.029591	0.006724	0.016001	0.013964	0.003449	0.000346
T45	0.002685	0.003628	0.000440	0.000683	0.001796	0.000187	0.001985	0.000498	0.000719	0.000346
T46	0.035840	0.032753	0.011936	0.002396	0.000882	0.031476	0.004006	0.037752	0.004053	0.010884
T47	0.015672	0.024372	0.027830	0.011195	0.025444	0.003304	0.096018	0.014215	0.096106	0.002253
T48	0.007870	0.033918	0.002255	0.026284	0.009030	0.047310	0.060478	0.027154	0.080483	0.000346
T49	0.064896	0.010311	0.001014	0.021361	0.025039	0.009885	0.028028	0.007273	0.010453	0.000346

Fig. 6. Artist and Topic Table

```
song_model.TOPICS[['Turnover','label']].sort_values('Turnover',ascending=False).head()
```

topic_id	term_str	Turnover	label
T37	0.140195	T37 heart, face, man, way, love, head, spit, g...
T26	0.113715	T26 oh, dance, love, life, cause, way, youre, ...
T47	0.096106	T47 life, light, tryna, eyes, day, dirty, pris...
T09	0.084188	T09 home, time, shots, mad, train, heart, gras...
T48	0.080483	T48 everybody, novel, dream, story, world, gir...

Fig. 7.

```
LIB_english.loc[top_match.index][['artist','song','top_genre']]
```

track.id	artist	song	top_genre
4LL5K6RPvbQz6CTcHvEwO0	The Seadogs	10,000 Miles Away	shanty
285pBltuF7vW8TeWk8hdRR	Juice WRLD	Lucid Dreams	chicago rap
31Gy7esHwiG8Cueoo944SQ	Turnover	Take My Head	alternative emo
5bwxMo2sye4bffTgp7uNbo	Turnover	Take My Head	alternative emo
4urxRqBRiaH0i20OKBsgxc	Three Days Grace	Break	alternative metal

Fig. 8.

A more sophisticated and mature implementation could filter out repeated songs and change the order if there are too many consecutive songs by the same artist. This can also extend to recommend songs outside of the library and encourage the user to explore new songs for their library.

Currently there are transformer models that can produce similar or better results for the task I just described through

semantic search and similar methods, however this is a method that requires much less computing resources.

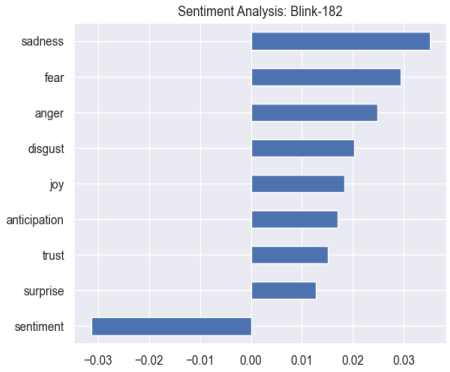

Fig. 9.

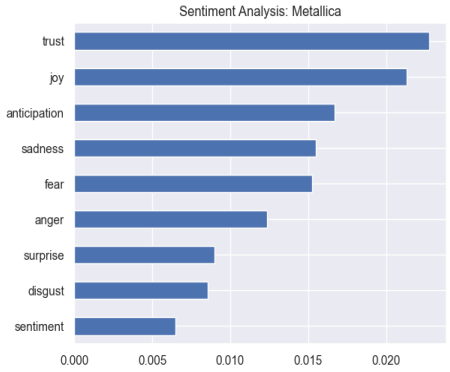

Fig. 10.

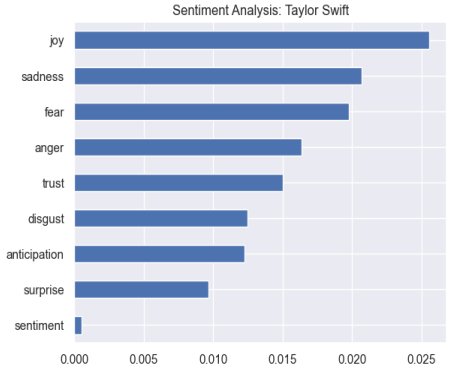

Fig. 11.

C. Sentiment Analysis

All of the steps highlighting our sentiment analysis can be found in the *7. Sentiment Analysis.ipynb* notebook in the root of this project directory. The lexicons were obtained from multiple sources can be found in the *lexicons* directory at the root of this projects repository. We combine all of the lexicon maps located in the lexicon directory in order to map them to our tokens table. The team then creates three tables within the notebook for sentiment analysis in order to separate the tokens for the top three most popular artists in the library. The sentiment vectors for a subset of emotions pre-defined for each of our top 3 artists are plotted in Figures 9 - 11.

From these plots we learn that from our predefined emotions/sentiments negative emotions are most associated with the lyrics of our Blink-182 songs. The rank of the distributions for these emotions seem to be a little more scattered for both Taylor Swift and Metallica songs in our library.

HCA for Cosine Distance Measure

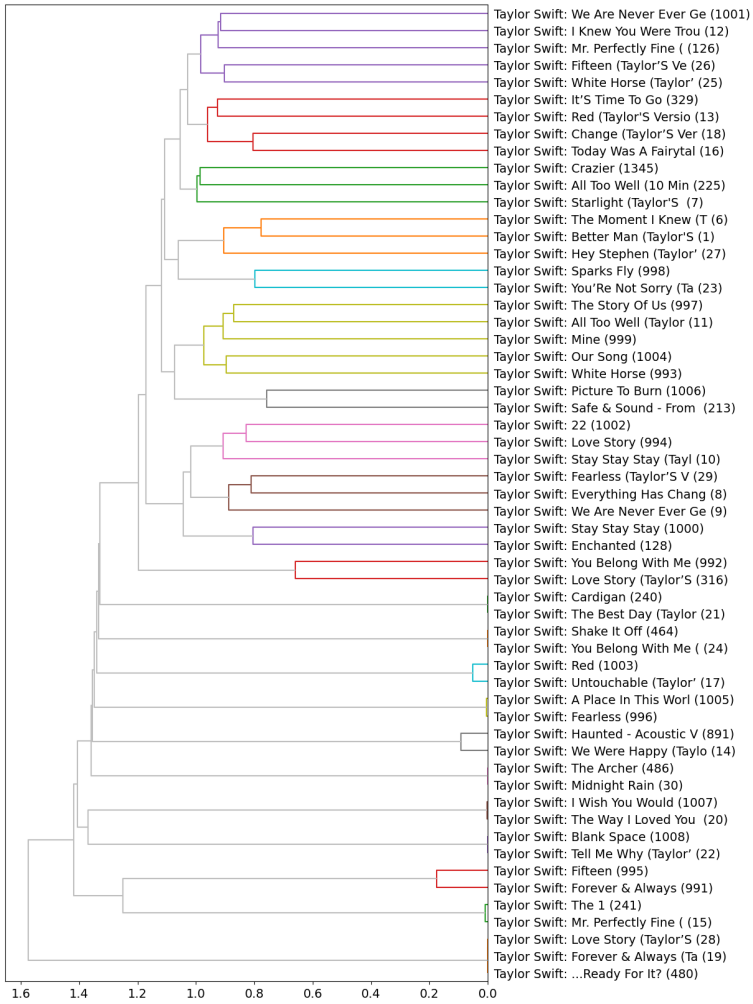

Fig. 12. Taylor Swift HCA

D. Hierarchical Cluster Analysis

Using the vectorized representations of our texts from the TFIDF table generated for our principal component analysis we are able to compute distances between individual songs and construct a hierarchical cluster plot them. We use cosine similarity as a distance measure and construct the hierarchy

cluster plot for Taylor Swift songs. This allows us to visualize and measure the similarity between the songs we have saved from any artist's discography. From Figure 12 we learn that *Forever & Always* and *...Ready For It?* were found to be close lyrically through our cosine distance measure.

IV. Conclusion

The multiple analysis approaches outlined above allowed us to gain a better insight into the content of any Spotify playlist or collection of songs from the platform. At multiple points decisions about the vocabularies had to be made, notably defining which parts of speech to keep for an analysis and whether or not to keep stop words. Removing stop words may help identify more distinct topics and better group bags of words for hierarchical cluster analysis, however, if one is comparing lyrical styles between genres, artists, or albums, it is important to keep stop words to capture writing styles. Lyrics provides a unique challenge since individual documents (songs) are shorter than books, magazines or articles. Additionally lyrics only captures half the story and the words of a song may not capture the entirety of its appeal. For example, *Take My Head* by the band Turnover has a rather high *valence* score on Spotify. Valence is a metric given to tracks by Spotify that measures their happiness or euphoric sound. The lyrics for this song is very dark and many casual music listeners may just enjoy songs for how they sound. This means that drawing conclusions from just one or the other may be misleading and future work should look at how to combine a lyrical analysis with that of sound characteristics.

APPENDIX

TABLE I
GENERATED TABLES

table	description
F1_CORPUS.csv	Contains the lyrics of every song in our Spotify library along with some metadata to identify the song
F1_CORPUS_english.csv	A subset of F1_CORPUS.csv containing only English songs.
F1_CORPUS_spanish.csv	A subset of F1_CORPUS.csv containing only Spanish songs.
https://app.box.com/s/g08v9rbz1wkj1dxhl1brdj6qh5a3danqF2_TOKENS_english.csv	Terms in our English Corpus indexed by our defined OHCO with POS, POS tuple, and term string as features
F3_top_artists_VOCAB_english.csv	Vocabulary of English songs with term string as index and term statistics as features including max_POS. This table is limited to only the top 10 artists form the English corpus.
F3_VOCAB_english.csv	Vocabulary of English songs with term string as index and term statistics as features including max_POS.
F4_TFIDF_english_small.csv	The TFIDF table for the top 1000 nouns and verbs in the English subset of our corpus.
F5_top_artists_DCM_english.csv	Statistics for the Document Classification Model using the F3_top_artists_VOCAB_english.csv subset.
F5_top_artists_LOADINGS_english.csv	The loadings calculated for the F3_top_artists_VOCAB_english.csv subset.
GENRES.csv	Genres and their appearance count across the entire corpus.
LIB.csv	The list of songs in our Spotify library and all of the metadata obtained from the API.
LIB_english.csv	A subset of LIB.csv only containing songs written in English.
LIB_spanish.csv	A subset of LIB.csv only containing songs written in Spanish

REFERENCES

[Danilak(2021)] Danilak, M. M. 2021, langdetect 1.0.9, https://pypi.org/project/langdetect/