Midterm Project

Nick Sun

2/10/2020

Introduction

Blink-182 was one of my favorite bands growing up and like many acts that have been around for a long period of time, their sound and lyricism has changed drastically over the years. I thought it might be interesting to apply some natural language processing tools on a corpus of all the Blink-182 lyrics I could find to see if I could quantify the ways they have evolved over time.

Coming into this project, I had 3 main questions which were all interrelated:

1. Are there certain topics that Blink-182 sings the most about and if so, how many topics are there?
2. Of the topics that were identified, what were some keywords or phrases that characterized that topic?
3. How has the lyrical content of Blink-182 changed over time?

Data Wrangling Process

The corpus of Blink-182 lyrics was scraped from the website `azlyrics.com` using the Python packages `requests` and `BeautifulSoup`. The advantage of using a fan-maintained site instead of a proprietary or commericial API is that the fan site contains transcriptions for live songs as well as demos and bootlegs that official sources might not have. The disadvantage to this approach is handling the nonstandard text descriptions that different users can provide, as well as dealing with various complications that are common web scraping.

The data wrangling process began with identifying the correct URLs to scrape from. This involved a preliminary scrape of Blink-182's artist page on `azlyrics.com` to get the links for all of the individual songs pages, contained as href attributes. I used a CSS selector query against the DOM to get a list of appropriate URLs.

Once I had a list of song URLs, the harder work began. First, I had to identify which CSS selectors would get the relevant information I wanted. Unfortunately, there were no ID attributes on any of the DOM elements, so I often had to resort to very specific CSS queries which relied on webpages having a very specific structure. There were several different webpage structures, depending on the information that the user community provided. Eventually after a decent number of tries, I came up with the appropriate queries to extract the information I needed from the webpages.

In all, I scraped lyrics, album name, and year of release from the individual webpages, resulting in a csv with three main columns, as well as additional columns storing the accompanying URL and the title of the song.

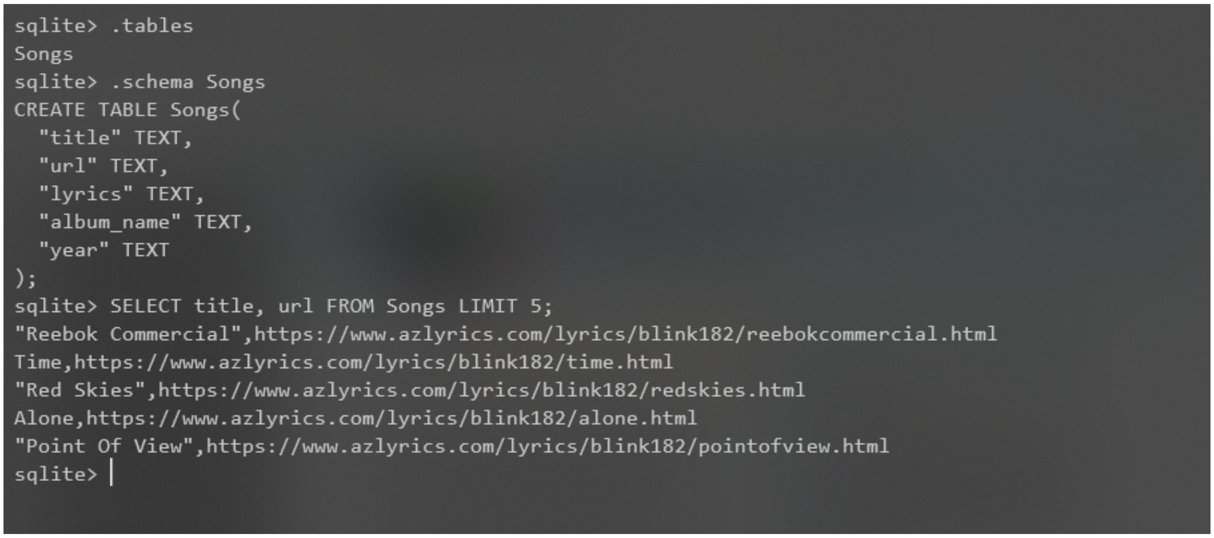
A final important key in web scraping was setting a sleep of a few seconds in between scrapes. Failure to do so will result in your traffic being blocked from the site. This unfortunately made the script slow to run, with the fastest I was able to run the entire script was 15 minutes against the entire corpus of lyrics.

Once I had the corpus scraped, I exported the resulting table into a csv. In the interest making the dataset more robust and shareable, I then converted the csv into a table in a sqlite database. I am a fan of doing things directly in the terminal, so I used the **sqlite3** command line utility to do this.

The exact commands I used to create a db with a single table called **Songs** were fairly simple:

```
$ sqlite3 blink.db
$ .mode csv
$ .import blink_songs.csv Songs
```

Once the tables were read it, I could execute queries within the sqlite shell. Below is an example of these queries, as well as the schema of the database.



```
sqlite> .tables
Songs
sqlite> .schema Songs
CREATE TABLE Songs(
  "title" TEXT,
  "url" TEXT,
  "lyrics" TEXT,
  "album_name" TEXT,
  "year" TEXT
);
sqlite> SELECT title, url FROM Songs LIMIT 5;
"Reebok Commercial",https://www.azlyrics.com/lyrics/blink182/reebokcommercial.html
Time,https://www.azlyrics.com/lyrics/blink182/time.html
"Red Skies",https://www.azlyrics.com/lyrics/blink182/redskies.html
Alone,https://www.azlyrics.com/lyrics/blink182/alone.html
"Point Of View",https://www.azlyrics.com/lyrics/blink182/pointofview.html
sqlite> |
```

Being able to execute queries in the terminal are one of the strong benefits of using sqlite over csv, even for a db which only has one table.

For use in a Jupyter notebook I wrote to explore the corpus using NLP tools, I created a database connection to the `blink.db` sqlite file. The code for this is relatively simple using the `pandas` library:

```
import pandas as pd
import sqlite3

conn = sqlite3.connect("blink.db")
df = pd.read_sql_query("select * from Songs;", conn)
```

The `df` object is now a Pandas dataframe containing the output of the SQL query, ready for analysis!

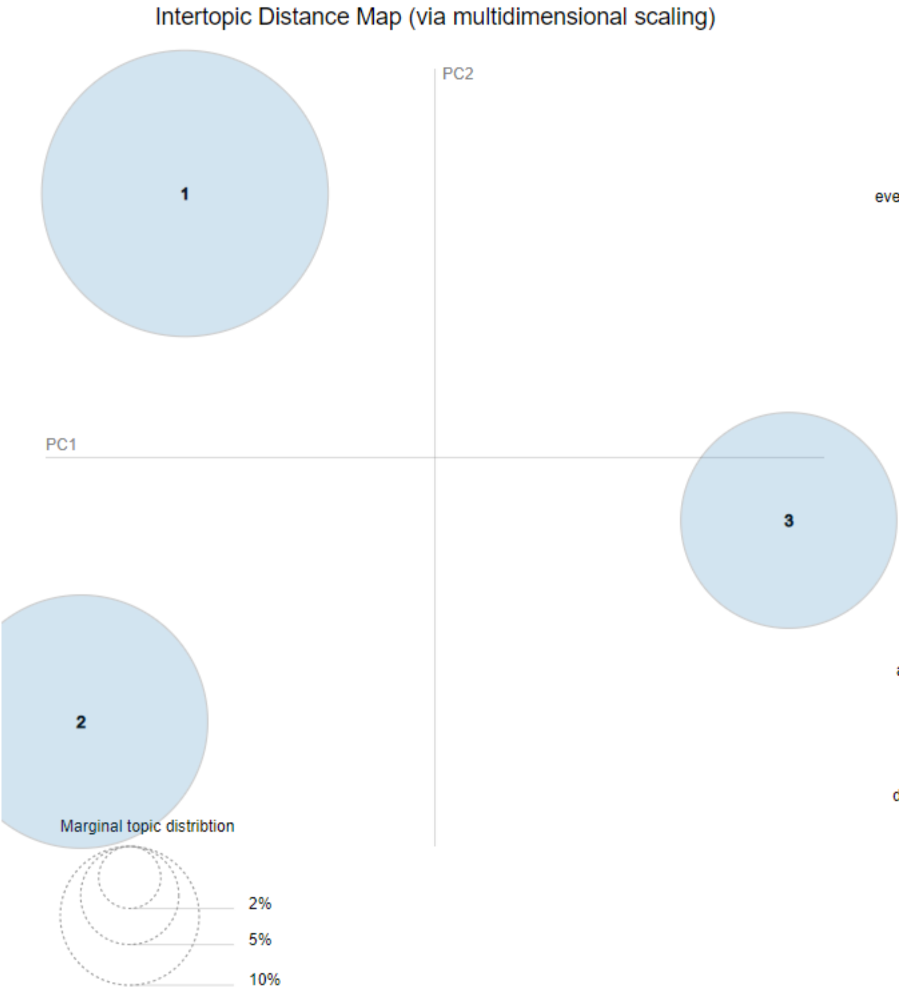
Analysis

I used the `nltk` package to further clean the text data. This includes stripping whitespace and punctuation, removing stopwords, and tokenizing the lyrics. Stopwords are generally considered as words that are unimportant to the semantic content of the document. A list of stopwords comes with `nltk`, but needed additional further tweaking since the user input found on `azlyrics` often had mispellings and fragments of words that had to be removed. Once I had a list of tokenized documents, I was finally ready to analyze text.

The tool I decided to rely on was **Latent Dirichlet Allocation**. LDA is a commonly used unsupervised learning technique that can ingest a corpus of documents and output a list of topics. The topics which are found by LDA are represented as a list of words that the algorithm deems to be of high importance.

I used the `gensim` and `pyLDAvis` packages to develop the LDA model and accompanying visualizations. Much like k-Means clustering, this technique requires that the user defines a fixed number of topics k everytime the algorithm runs. After trying different values of k from 20 to 3, I noticed that even when using a high number of topics, the topics clustered together. Therefore, I opted to collapse the topics and keep $k = 3$, meaning that there were **three main topics** in the corpus of Blink-182 lyrics.

This is visualized in `pyLDAvis` using multidimensional scaling which is a technique similar to PCA which reduces high dimensional data like text to lower dimensions where they can be represented. Each of the three large blue cirlces indicates a cluster of documents belonging to a single topic. Again, we can see that there are three distinct topics, according to the LDA algorithm.



The most important, or as LDA documentation terms it **salient**, terms can be found again using the `pyLDAvis` packages which provides an interactive barchart for exploring the most important terms in the entire corpus and in the individual clusters.

In the table above, we can see the most frequent words in the entire corpus. As one might expect, some of the most popular words are nonsense words like “na”, “oh”, “la”, and “woo”. Furthermore, exploration of these words shows that the frequency of these words is pretty constant throughout time. The saturation of these words adds to the difficulty of determining distinct topics in the corpus.

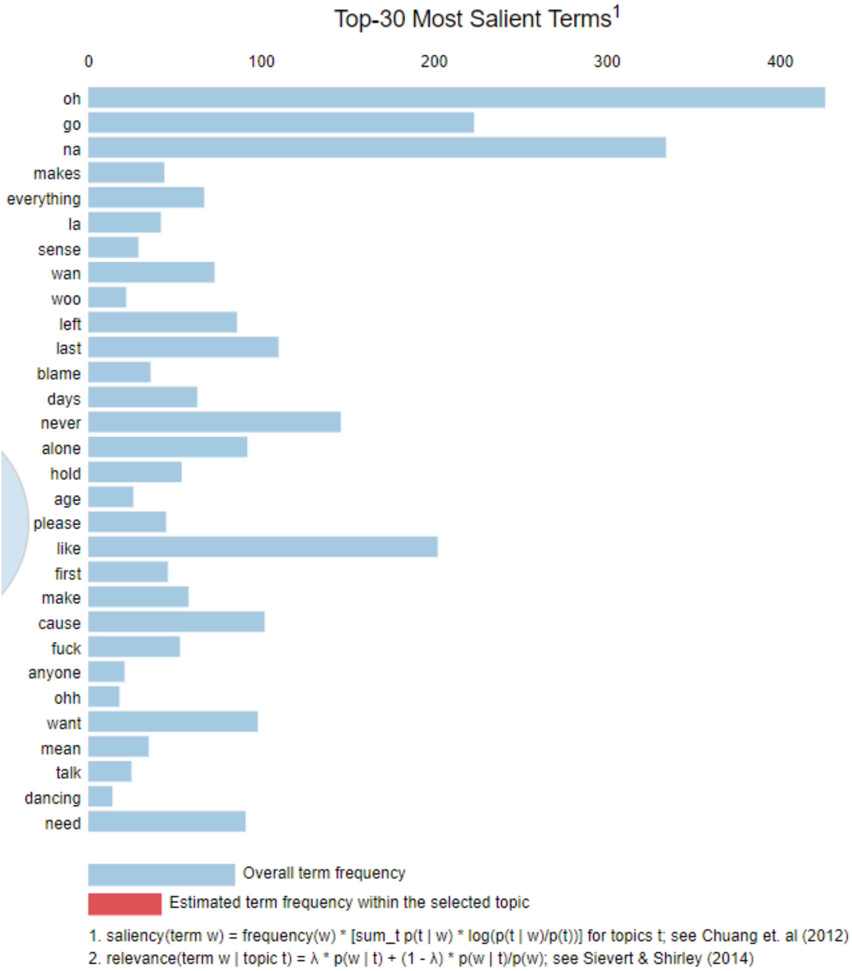
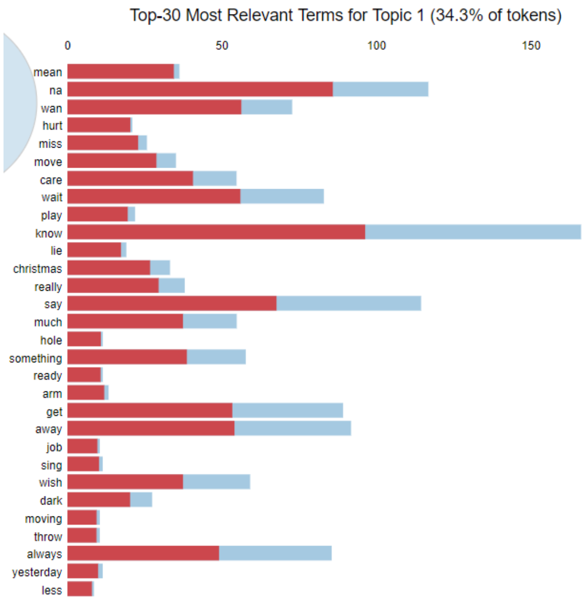
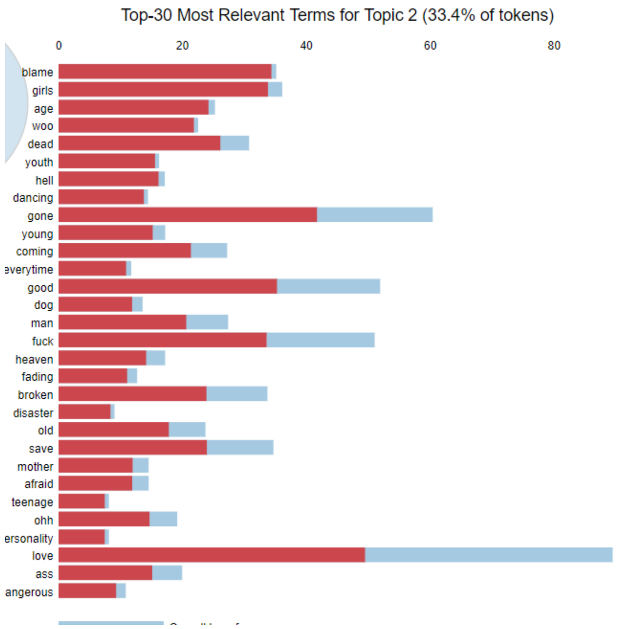
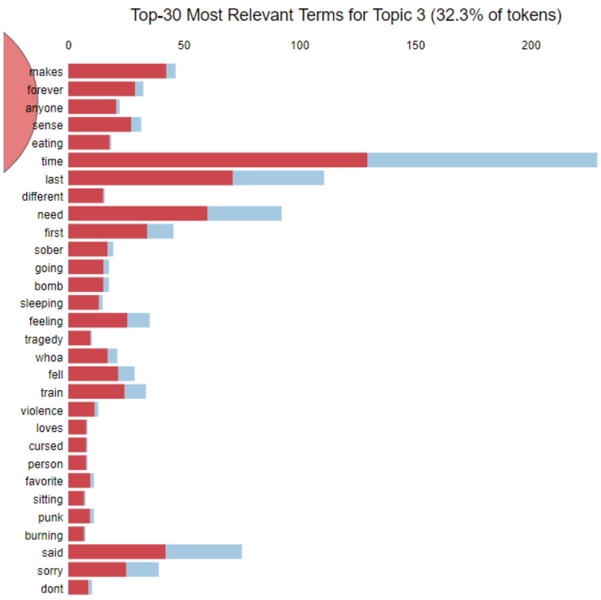


Figure 1: Top 30 most important words in the corpus as per LDA





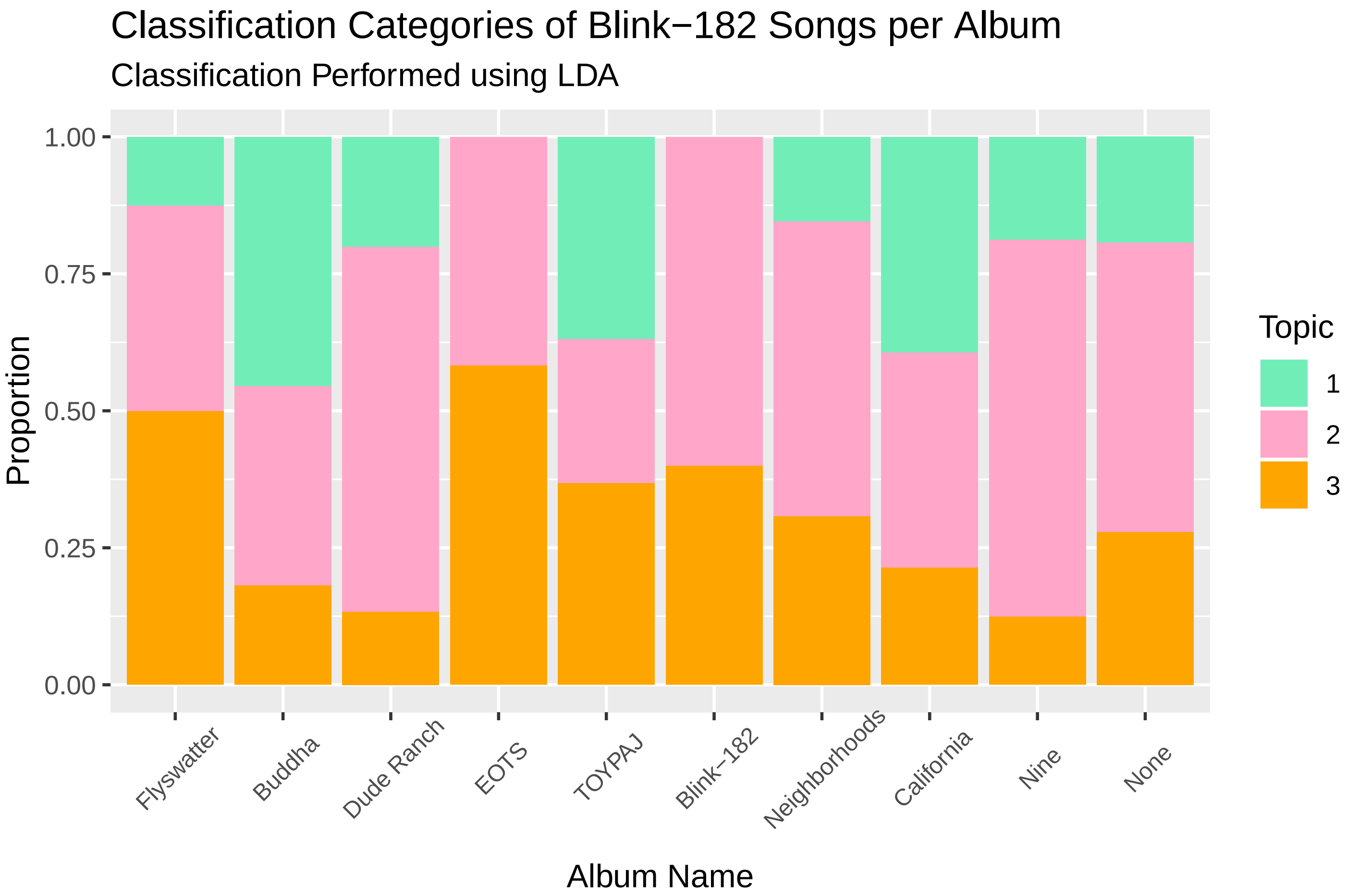


We can identify the main content of the topics by looking at the most important words contained in each topic. In the above barplots, the blue color represents how common that word was in the overall corpus and the red represents how many instances of that particular word were found in that topic. After playing with the tuning parameters of this model for a while, it became clear that the majority of Blink's catalog uses similar vocabulary (think "hey", "na", "whoa" kind of vocabulary) which made it difficult to get clear clusters. Tightening the parameters to only identify words special to each topic was necessary to get an idea of what the topics were. This exercise is more art than science, but from the words LDA identified, it appears that we have the following rough topics which answer our second question of interest:

1. Topic 1 is basic skate punk with lots of chant-along lines like whoas, na nas, etc.
2. Topic 2 is songs about youthfulness (see "young", "teenager", "age", "youth", etc.) and girls
3. Topic 3 contains songs that seem to have more serious or complex subject matter (see "bomb", "violence", "forever", "sober" "tragedy")

For our last question, I am interested in any systematic variation between albums and their topical content. I hypothesized from the start that we would see a pattern of Blink's songs overall getting more mature as time went on (although their recent albums still included some juvenile humore).

I explorted cleaned LDA data into `R` for easy visualizing using `ggplot2`. I decided to visualize the changing proportions of each of the three topics we identified using a stacked proportional bar chart.



Classification Categories of Blink−182 Songs per Album

Classification Performed using LDA

The 'None' album label refers to live songs or improvisations.

Conclusions

Our LDA algorithm found three main topics within our corpus of Blink-182 lyrics. These topics were loosely characterized as the following:

- One song topic was heavily based around nonsense words and chant-along choruses
- Another topic based around youth, such as dating girls and other teenage activities
- The last topic was centered around more serious content, such as death and violence

Songs can be about multiple topics, but each song has a predominant topic which is identified by a LDA allocated weight. Using these weights, we can simplify our analysis characterize each song as belonging to one topic.

Finally, our last question asked if we could identify a systematic trend in the lyrical content of Blink lyrics over time. We can answer our final question of interest using the above barplot where albums are plotted chronologically and different colors represent proportions of songs belonging to a particular topic.

This visualization is actually somewhat surprising. Topic 3 characterizes the most mature lyrics, but it seems that the album with the most songs from this topic is mid-career, peak Blink-182 with the album Enema of the State. The other album that approaches it in maturity is the self-titled 2005 release which was a decidedly darker direction which fits with the album having more serious and mature songs. In particular, the album really deviated stylistically from earlier albums which made it a little controversial with fans when it first came out. This change is evident in the lyrics, as self-title album is much different than the prior album Take Off Your Pants and Jacket, which as the name might imply made heavy use of juvenile themes and sometimes immature language, especially on the extended cut.

Another interesting thing is that the lyrical maturity of Blink seems to be going back to its original roots. Topic 1 actually makes a significant chunk of new Blink songs from the album California onwards. I did get the impression upon listening to these albums that Blink was trying to recapture their more youthful energy and punk rock roots (even though they are 40 somethings now and can't really sing about immature topics without seeming hammy). The album Nine having a strong emphasis on youthfulness makes total sense to me as a significant theme of the album is nostalgia. This theme becomes very apparent from the first track which looks back fondly on youthful mistakes ("there ain't nothing like the first time").

Maybe if Blink stick around for another 10 years, we can check if this pattern is actually cyclical and they will go back to singing about mature topics like death and taxes.