Music discovery through song lyrics

Project LYRICAL - Mateus Surrage Reis, Indrek Ardel

Introduction
Music is an area of wide human interest, and as such, generates a proportionally large amount of raw data in today's connected world. The abundance of music often makes it challenging to discover or replay new songs that a person might enjoy. In the project, we sought to find how songs are connected with each other through their lyrics and how these methods could be used to find similar music.

Challenges
Our datasets were drawn from a variety of online sources, and those sources in turn drew from myriad sources, minimally curated and moderated output. This, along with other factors inherent to using online sources, song lyrics, and text in general as data brought with it a unique set of challenges when it came to data preparation.

First and foremost is simplicity, not in the lyrics, but in attached information such as year of release and genre, after all, just lyrics by themselves are typically not that interesting. To this issue there's no good solution but to find better databases, or scrape and compose some yourself. Due to the lack of quality year information, we had to abstain any plans of providing analysis that would have compared songs by year.

We also had to take into consideration that songs come in many languages, and this was in fact the most complicatedly expensive section of pre-processing, as we relied on external libraries for the task of removing non-english song.

Results, there are more common diacritics in such databases (many artists are attributed to same song even if they aren't the original author), encoding issues, and unwritten artifacts in data. As an example, lyrics text often contains indicators for chorus, verses, or how many times a part of the song is sung. Those were filtered as they were indeed the normal way.

Data
We used various sources for obtaining lyrics. Datasets were discovered through Kaggle. All data and scripts used to process data is available at our project repository[1].

1. Billboard Year-End Hot 100 charts from 1959-2018
   This is an embedded version of the dataset provided by a Github user Kayin-Park[2], which contains data from Wikipedia, lyricsmode.com, genius.com, songlyric.com and metrolyrics.com.
2. 500 000+ songs from lyrics.com
   We scraped the website using the sources provided on its corresponding Kaggleg page in hopes of getting more and most recent songs, however out of 1.3 million records we still found only around 500 000 to have unique lyrics.
3. 380 000+ lyrics from Metrolyrics
   This dataset was sourced from Kaggle, it is worth mentioning that this dataset includes records we still found only around 500 000 to have unique lyrics.
4. 35 000+ songs from LyricsFreak
   We re-scraped the website using the sources provided on its corresponding site hoping to get more and more recent songs.

Make your own playlist
In order to explore the large number of songs and in which way they are similar to the Billboard Year-End Hot 100 chart songs, we crafted a web application which you can try out by scanning the QR code found on the poster.

Finding similar songs based on word sequences
In addition to log-likelihood, another method of discovering songs with similar themes is through matching word sequences found in lyrics. If songs share a significant amount of word sequences, they can be thought to be similar.

The word sequences used in this project are simple word n-grams of size 3 in order to strike a balance between the amount of tokens generated while still having the ability to convey some meaning for songs written in English.

Songs in dataset 1 (Billboard Year-End Hot 100 charts from 1959-2018) were tokenized and paired up with the rank of songs in dataset 2 and 3, which contain many new songs compared to what we already had in datasets 2 and 3, which

Country

Finding similar songs based on word sequences
In addition to log-likelihood, another method of discovering songs with similar themes is through matching word sequences found in lyrics. If songs share a significant amount of word sequences, they can be thought to be similar.

The word sequences used in this project are simple word n-grams of size 3 in order to strike a balance between the amount of tokens generated while still having the ability to convey some meaning for songs written in English.

Songs in dataset 1 (Billboard Year-End Hot 100 charts from 1959-2018) were tokenized and paired up with the rank of songs in dataset 2 and 3, which contain many new songs compared to what we already had in datasets 2 and 3, which

How did we reach these words?
The song sequences we used, as well as the highlighted rows were the ones with highest log-likelihood of appearing in their respective genres, calculated with the following formula:

\[
\log \text{likelihood}(\text{song}) = \log \text{likelihood}(\text{song sequence}) = \sum \log \text{likelihood}(\text{word sequence})
\]

Overview
- \( \text{O}_i \) is the amount of occurrences of that word in that genre's lyrics.
- \( \text{F}_i \) is the expected amount of occurrences of that word in the genre's lyrics, if it had been uniformly distributed across the lyrics.
- \( \text{E}_i \) is the expected occurrences in all other genres, as above.

Finally, the log-likelihood was ranked if \( \text{O}_i > \text{E}_i \) which is greater than \( \text{O}_i \).

How did we reach these words?
The words in the tables, as well as the highlighted rows were the ones with highest log-likelihood of appearing in their respective genres, calculated with the following formula:

\[
\log \text{likelihood}(\text{song}) = \log \text{likelihood}(\text{song sequence}) = \sum \log \text{likelihood}(\text{word sequence})
\]

Overview
- \( \text{O}_i \) is the amount of occurrences of that word in that genre's lyrics.
- \( \text{F}_i \) is the expected amount of occurrences of that word in the genre's lyrics, if it had been uniformly distributed across the lyrics.
- \( \text{E}_i \) is the expected occurrences in all other genres, as above.

Finally, the log-likelihood was ranked if \( \text{O}_i > \text{E}_i \) which is greater than \( \text{O}_i \).

The remaining words in the tables were discovered by going through the lyrics again and extracting all words within two words of the highest-log-likelihood one, then ranking them by absolute number of appearances. The columns in those tables are those that are selected from amongst the top-ranked words. That is to say, in the columns are words that appeared frequently together with the entry at the top of the column.

References
1. Project LYRICAL repository
2. https://github.com/Surprise1999/Project-Lyrical
3. Amy White - How to Strike a Match
4. Simon White - How to Strike a Match
http://www.catalysoft.com/articles/StrikeAMatch.html
https://github.com/walkerkq/musiclyrics