





# Project iMuse: an Interactive Visualizer of Lyrical Sentiment

A. Lu<sup>1</sup>  and J. Anstey<sup>1</sup>  and Z. Zhang<sup>1</sup>  and R. Wang<sup>1</sup> 

<sup>1</sup>Emory University, United States of America

---

## Abstract

*Our interactive visualization, Project iMuse [SA22a], provides the unique ability to view the most used words in popular music over the past 50+ years in conjunction with how they were used in songs through sentiment analysis. To aid in more detailed analyses, Project iMuse has the ability to dynamically consider a variety of user defined subsets. These subsets are created through the user interaction, which includes changing the range of years considered and selecting particular word types.*

## CCS Concepts

• *Human-centered computing* → *Information visualization*; • *Applied computing* → *Arts and humanities*;

---

## 1. Introduction

As a part of an undergraduate computer science elective entitled “Information Visualization,” we were tasked with a multi-month long final project resulting in an interactive visualization. Our group chose to work with music since there were swathes of public data available and because we are passionate about it: multiple group members are musicians and/or fans of music itself.

### 1.1. Research Questions

With the prompt and variety of data sources available to us, our research led us to considering questions regarding lyrics. Specifically, what are the most commonly used words in popular songs? Following this core research question, we began to ponder related queries: what type of words are used in popular songs? Does a word’s popularity change over time? In what sentiment are these words used in songs; do they have a positive or negative connotation? Our interactive visualization was then constructed with the explicit purpose to answer those questions.

## 2. Related Work

A YouTube video published by Vox Media, “Why we really really like repetition in music,” [Cas17] is arguably the highest viewed lyric visualization ever. With nearly 2 million views it showcases a few tools and sub-sources that demonstrate a variety of ways to visualize and interpret lyrics.

The first was a highly stylized collection of SongSim [Mor17a] matrices. The neatly arranged grids of neon-lit squares represented pieces of songs that would repeat themselves. Morris, the creator of SongSim, was interviewed in the video where he provided how SongSim matrices are generated. Each row and each column in the matrix represents a lyric in the song and the matrix is filled in

when a row and column match. Clusters of points then show general lyrical repetitiveness and the density of these clusters indicate how repetitive the lyrics are at a local level.

SongSim was not the lyrics related visualization Morris had made either. Later in the same video, Vox displayed a graph from Morris’s article “Are Pop Lyrics Getting More Repetitive?” [Mor17b]. In this article, Morris takes a unique approach to lyrical analysis through taking a song’s lyrics and then compressing them. The greater the compression, the greater the repetition. The article then creates bar charts showcasing different well known songs and how their repetitiveness has gone up over time.

While we were unable to find anything that was very similar to our interactive visualization, this collection of sources provided the necessary background to approach this problem in a manner where we were set up for success.

## 3. Dataset

### 3.1. Finding the data

There were no existing datasets that fit our needs of having the lyrics of popular music, so we needed to create our own. We started by finding a dataset which would provide us with what songs were popular over time. From there, we could gather the lyrics ourselves to then prep for analysis. Our work then began with a dataset from kaggle. Entitled “Billboard ‘The Hot 100’ Songs,” this dataset included a monthly snapshot of the Billboard Hot 100, a service that catalogues the most popular music at any given time, from 1958 to 2021 [Dav21]. Using this dataset as a base, we could then begin to get popular song lyrics.

### 3.2. Data collection

After some basic cleaning, we created a python program [SA22b] to create the exact dataset we were looking for. This included getting the exact year a song was published through the Spotify API [Spo23] and the lyrics to said song through Genius [Gen23].

From the initial 29,680 unique songs found in the original kaggle dataset, our methods of data collection were able to gather the lyrics to 23,195 of them. We believe the missing lyrics are a combination of a few factors. Namely, Genius URLs not having a consistent format, the song being an instrumental, or Genius not having the lyrics at all.

After this process of gathering and further cleaning, we were left with a dataset containing two columns: the year a song was published and its lyrics. For our interactive visualization to perform, we needed to further deconstruct this data.

## 4. Data Wrangling and Machine Learning Methods

### 4.1. Lyrics Cleaning

To ensure the validity and dependability of our data analysis, we initially conducted a process recognized as "lyrics cleaning." The procedure comprised several steps, beginning with the removal of non-English letters other than apostrophes, alongside punctuation, special characters, and non-Latin characters to create a standardized dataset. Subsequently, we eliminated stopwords, which are words that are irrelevant for sentiment analysis, utilizing a list furnished by the Natural Language Toolkit (NLTK) library [NLT23a]. We then proceeded to expand all contractions such as "you're" to their full form, such as "you are," and lemmatizing the remaining words with Part-of-Speech (POS) tags utilizing NLTK to maintain consistency in form. Once the lyrics were subjected to the cleaning process, we proceeded to compute the frequency of unique words found.

### 4.2. Sentiment Analysis

Following the process of cleaning the lyrics and computing their frequency, we proceeded to conduct a sentiment analysis using a Valence Aware Dictionary and sEntiment Reasoner (VADER) [NLT23b] tool, which assigns a compound sentiment score ranging from -1 to 1, with -1 indicating very negative sentiment and 1 indicating very positive sentiment. We would first calculate the sentiment score of each song and assign the score to each word in the song. To derive an overall sentiment score for a given year, we calculated the weighted average sentiment score by summing the product of each word's frequency score and sentiment score, and dividing this sum by the total frequency of all words within that year. This method enabled us to obtain a comprehensive understanding of the sentiment trends in the corpus of lyrics we analyzed.

## 5. Example Findings

Depending on the manner in which the webpage is used, several hypothesis can be answered, including our research questions. Potential findings include the observation of a discernible negative

trend in the lyrical content of songs over the years. Moreover, a comparative analysis demonstrates that the sentiment polarity of language in older music exhibits greater contrast in comparison to that of more recent music. Additionally, a significant trend in the popularity of colors used in lyrics is the prominence of the color "Blue" throughout the years. These potential findings suggest that the webpage could be a valuable tool in providing insights into the evolution of language, cultural trends, and the use of color in popular music.

## 6. Reflection and Future Work

The analysis of language requires consideration of its contextual nuances. To ensure consistency in our analyses, lemmatization is employed to treat words with similar meanings as the same word, despite their different forms. However, as words can have multiple lemmas, the introduction of a POS tag is necessary to accurately reflect the context of the word, albeit with the caveat that this approach may not always be infallible.

In the case of songs, certain words may be censored or obscured with symbols such as asterisks, which necessitates their exclusion from analyses.

Some potential improvements for the system include the following:

- Search individual words: The system could be improved to allow for searching individual words, which could enhance the accuracy and specificity of search results.
- Input exact number of bubbles: Currently, users can only specify a general range for the number of bubbles displayed. Allowing users to input an exact number of bubbles would provide more precise control over the visual display.
- Scalable bubbles: To ensure optimal display on screens of different sizes and resolutions, the system could be improved to make the size of the bubbles scalable according to the screen resolution. This would help to maintain the relative size and spacing of the bubbles, regardless of the device being used.
- Consistent frequency to size ratio: In order to avoid visual distortion and ensure the accuracy of the data representation, the system could be improved to maintain a consistent frequency to size ratio for the bubbles. This would ensure that the size of the bubbles accurately reflects the underlying data, and facilitate comparisons between different bubbles.

## 7. Conclusion

In conclusion, this research presents an interactive visualization tool, Project iMuse, which offers a novel way of analyzing and understanding the use of language in popular music. Unlike traditional lyrical visualizations that focus on word frequency alone, Project iMuse integrates sentiment analysis to provide a more nuanced view of the lyrical content. By allowing users to explore subsets of popular words based on different criteria such as year range or word type, Project iMuse offers a more customizable and user-friendly experience. Overall, this tool has the potential to provide valuable insights into the evolution of language and cultural trends in popular music, both past and present.

## References

- [Cas17] CASWELL E.: Why we really really really like repetition in music, Oct 2017. URL: <https://www.youtube.com/watch?v=HzzmqUoQobc>. 1
- [Dav21] DAVE D.: Billboard "the hot 100" songs, 2021. URL: <https://www.kaggle.com/ds/1211465>, doi:10.34740/KAGGLE/DS/1211465. 1
- [Gen23] Genius, 2023. URL: <https://genius.com/>. 2
- [Mor17a] MORRIS C.: About, Feb 2017. URL: <https://colinmorris.github.io/SongSim/#/about>. 1
- [Mor17b] MORRIS C.: Are pop lyrics getting more repetitive?, May 2017. URL: <https://pudding.cool/2017/05/song-repetition/>. 1
- [NLT23a] Nltk documentation, Jan 2023. URL: <https://www.nltk.org/>. 2
- [NLT23b] nltk.sentiment package, Jan 2023. URL: <https://www.nltk.org/api/nltk.sentiment.html#module-nltk.sentiment.vader>. 2
- [SA22a] SUBMITTERS, ANONYMOUS: Project imuse, Dec 2022. URL: <https://jack-anstey.github.io>. 1
- [SA22b] SUBMITTERS, ANONYMOUS: Spotify-scraper, Sep 2022. URL: <https://github.com/Jack-Anstey/Spotify-Scraper>. 2
- [Spo23] Spotify web api, 2023. URL: <https://developer.spotify.com/documentation/web-api>. 2