Exploring Censorship and Lyric Alterations in Kidz Bopz Music from 2001-2019

Salinas, Nicholas
Department of Electrical Engineering and Computer Science
Massachusetts Institute of Technology
Cambridge, Massachusetts
nsalinas@mit.edu

Wu, Shannen
Department of Electrical Engineering and Computer Science
Massachusetts Institute of Technology
Cambridge, Massachusetts
shannen@mit.edu

Yin, Jessica
Department of Electrical Engineering and Computer Science
Massachusetts Institute of Technology
Cambridge, Massachusetts
yinj@mit.edu

***Abstract*—Censorship and lyric alterations in music is observed in many public media channels such as the radio, TV, and film among others. Often, it's easy to identify certain profane words and find a replace the word in the lyric to produce a "Clean" or "Radio Edit" of a song. However, in the case study of Kidz Bop, a brand known for making songs as kid friendly as possible, we can visualize these trends to understand how music is altered for young audience, and in some cases, can change the context of a lyric or complete remove it. Our interactive data visualization tools process a hierarchical data transformation to create trends and associations among songs, artists, and altered words found in our music.**

Keywords—Kidz Bop, React.js, D3, Bubble Visualization, Treemap, Music, Songs, Lyrics, Censorship, Hierarchical Data Transformations

I. Background & Introduction

The motivation of this project is to develop creative and interactive data visualization tools to understand how music and lyrics are altered, censored, or both by Kidz Bop, and to identify insights and trends found in the music. Our robust data set provides plenty of information, and our combined approach with React.js and D3 produces two new visualizations to this problem. While Kidz Bop is a small case study of the music industry, we hope this project can be extended to provide insight on how radio, TV, film and other media channels approach censorship or lyric alterations in music.

II. Related Work

During our project development, we have not been able to identify any robust visualizations focused on lyric analysis. We used prior examples from the D3 community to understand how to build certain interactions and animations, and we tailored our visualizations to work in unison with our hierarchical data set.

III. Methods

A. About our Data Set

We encountered a public data setting provided by Pudding that consisted of more than 1300 entries. The following data fields as defined:

- `badWord` – the bad word or phrase from the original lyric that was censored or altered
- `category` – the type of censorship that this word falls under six nominal categories
- `count` – the unique counter that any lyric alteration occurred in the song
- `ogArtist` – the original artist(s) of the song
- `ogLyric` – the original lyric found in the song
- `kbLyric` – the new lyric written by Kidz Bop. The data table indicates "cuts verse" if the lyric is removed entirely from the song.
- `songName` – the name of the song
- `year` – the year the song was published

B. Transformations with D3

We used D3's data transformation tools to create a nested hierarchical data structure. Because our original data set had multiple nominal fields such as artist, song name, category of censorship, and bad word, we could design our own hierarchy to group our dataset.

Our starter bubble visualization used the following ordering mechanism to group our data: (1) Artist, (2) Bad word, (3) Song Name. All the "leaves" of our nested data structure represent the original data points, and this nested data structure allows our audience to use a depth-first exploratative approach to our data.

We conducted a similar hierarchical data transformation for our treemap visualization. However, we decided to incorporate temporal data, and group our data using the following order: (1) Year, (2) Category, (3) Badword. This short and simple ordering mechanism meant that many leaves would be grouped together by the third level of our nested data structure, and this allowed us to use modals to have the user explore and analyze a lot more data in one view. Using the category field also allowed us to use our color encoding channel to understand how different sectors in language were altered over time.

IV. Interactive Bubble Visualization

A. Packing our Data

Using D3's Hierarchy library, one tool that we used was `d3.pack()`, which allowed us to feed in our hierarchical data structure to D3, and it would return a list of nodes, each with their own data and respective children. This approach also provided us a coordinate position and radius for each node, and allowed us to represent our data as circles such that no two circles would ever overlap.

B. Artist Typeahed Search Bar and Dropdown Menu

To allow users to specifically search for an artists' songs, we implemented a dropdown menu above the bubble visualization. Users can browse by scrolling through the artists in the dropdown or typing the artist's name which will filter the dropdown options. We chose to include a dropdown menu so all artist names could be visible, as some names were too long to explicitly fit and show in each bubble.

C. Bubble Radius & Size Encoding Channel

We used a size encoding channel to represent the number of songs each artist had words censored in. A larger bubble corresponds to an artist with more songs censored. Using the bubble size was the best way to compare across different artists as differences in size was more easily visualized.

D. Nominal Lyric Categories & Color Encoding Channel

Each censored word in the dataset was classified in one of the following nominal categories: alcohol and drugs, sexual, profanity, identity, other, and violence. In the bubble visualization, we used a color encoding channel to indicate the category a censored word falls in. A legend is visible on the words layer and song layer to quickly associate the color with category. The legend disappears on the outermost layer to make it clear that the colors of those bubbles are not associated with any category. This combination of color and size encoding makes it easier for a user to identify which category of censored word an artist uses most often in their music.

E. Labels & Text Encoding Channels

On the outermost layer we label each bubble with an artist name. We omit showing the artist's name on a bubble if the name overflows the bubble's circle border. To do this, we check the number of characters in the artist's name against the D3-defined radius of the bubble. If the number of characters is less than the diameter of the bubble, the artist's name is omitted, and an on-hover tooltip is used instead to display the entire name. Through many iterations of testing, we found that using the bubble's diameter was the best measure to determine whether the artist label will overflow the bubble or not. Only showing artist names that fit in their bubbles also has the added benefit of decluttering the visualization, making it easier to read and interact with.

F. Side Navigation Menu

Given the number of layers in the bubble visualization, it can be easy for a user to be confused or lose track of where they are in the visualization. Thus, we created a sidebar that shows all the layers in the depth-based visualization, what layer they are on, and what path they have selected in previous layers. This sidebar updates as the user explores the visualization, giving more context to the bubbles they selected.

G. Interactive Lyric Comparison Modal

To allow our audience to fully interact with the data points, we decided to incorporate interactive modals. The modal for the bubble visualization was our first prototype and attempt at displaying all the data. Specifically, we wanted to display all the data that corresponded with a certain "leaf-level" node, meaning that this node had no children data, and represented actual data entries.

This modal contains the comparison between the Kidz Bop lyric and the original lyric, and we decided to include these two pieces of information in the modal as they were the most representative and requested pieces of information that made sense for our visualization and lyric analysis. We continued to iterate on this design and improved upon it for our Treemap visualization.

V. Temporal-based Treemap Visualization

To improve upon our visualization, we decided to build a treemap, a new and often underused visualization technique, to show word relationships. Our hierarchical data structure was designed perfectly to build our treemap visualization. We also decided to use more of our dataset by incorporating the temporal year field from each entry.

Our treemap visualization uses a different hierarchy to order and group our data: (1) Year, (2) Category, (3) Bad word.

A. Use of Time Scale to Filter by Year

Our treemap displays censored words across all Kidz Bop songs for a specific year, or all years. The user can filter by year with our year menu on the right or choose to display the data from all the years combined. This allows users to view how each word or grouped category has changed in censorship over the history of Kidz Bop.

B. Representing Most Altered Words with Size Encoding Channel

We used the size encoding to represent the count of words censored in songs for a given year. The larger the tile, the more times that specific word is censored each year. Placing different sized tiles side by side in a treemap visualization makes it easy for users to compare different word counts against each other. The size of each tile is solely dependent on its count and is not related to any other fields in our data such as artists or song name.

C. *Hiding Profane Words with External Filtering Library*

We included a third-party profanity filter that can be toggled on and off with a checkbox. We understand that there will be users who are uncomfortable viewing so much profanity on the screen at a time, so we incorporated this filter to let users more comfortably explore the data. The default setting of our filter will automatically hide any profane words when the visualization is first opened.

D. *Interactive Music Data Modal*

Similar to the bubble visualization, we decided to iterate and improve our interactive modal design that allowed the user to click on the artist in a menu section on the left. The content on the right will then update as each artist menu item is clicked. The lyrics are grouped by the song name and displayed in a similar format to the bubble visualization. We used consistent coloring with the legend and Bubble visualization in the modal to highlight which category a certain word has been altered by.

It's important to note that the lyric comparisons in the modal from each song only show those that come from the word tile that the user first clicked on. Showing a certain song in the modal may not reflect all the lyrical alterations in that song.

E. *Represented Words by Nominal Category using Color Encoding Channel*

In the treemap visualization, we used a color encoding channel to indicate the nominal category a censored word falls in. We used the same colors as in the bubble visualization for consistency. Beside the treemap, we display a color legend sorted by the largest category of censored words. For each category in the legend, we also included a count of the number of censored lyrics for that category in the selected year. This allows users to quickly associate the colors with their categories and see which category of censored words was the most popular in a given year. The combination of grouping of censored words by category, color encoding, and size encoding channels makes it easier for a user to see how different categories of censored lyrics compare and change over time.

VI. Interaction & Animation

As part of our motivation, we wanted to incorporate interaction and animations into our visualizations not only to provide a better user-friendly experience, but to also allow the user to discover and infer their own insights about the Kidz Bop music.

A. *Zoomable Bubble Visualization*

One of the interactive features we were particularly drawn to is the automatic zooming of the visualization when a user clicks on a certain bubble. Fortunately for us, D3 came with a pre-built hierarchy library that generated the bubble's coordinate position and radius, and we decided to update the focus of our visualization based on user clicks. This depth-first discovery and interaction allows the user to hit the "Escape" key to zoom out one level at a time of their visualization.

B. *Interactive Year Menu*

On the right side of our treemap visualization, we display an interactive year menu. Users can click "All years", or a specific year to filter the data shown in the tree map. This allows users to easily explore our dataset and see how Kidz Bop censorship has changed over time.

C. *Hover Animations*

For both visualizations, we created hover animations to reveal features that are interactable. In the bubble visualization, we add a glow shadow effect over the bubble borders that are triggered when the user hovers over each bubble.

In the treemap visualization, we add a heavy white border hover effect around the box for a better user experience. Additionally, both visualizations utilize tooltips that provide further context on what the bubbles/boxes mean. In the bubble visualization, a tooltip will appear on-hover over a bubble that is too small of a radius to display the full artist's name, providing an elegant way of labeling bubbles without clutter.

The tooltip in our treemap also appears over each box to further describe what bad word and count the text and size encoding correlates to. Like the bubble visualization, the tooltip provides a way to label smaller boxes that do not have the space to display a full word.

D. *Displaying and Hiding Interactive Modals*

For both visualizations, we implemented an escape key event listener that allows for easier opening and closing modal displays. If a modal is currently open, pressing the escape key will close the modal. In the bubble visualization, pressing the escape key will zoom the user out to the previous layer. A small instruction is fixed to the top right of the screen to make users aware of this navigation tip when navigating the bubble visualization.

VII. Exploratory Data Analysis

When looking at our data set and visualizations, we really used our size encoding channel to isolate and identify certain trends and associations. Kidz Bop as a brand tends to cover songs on the Top 40s chart from major pop artists such as Ariana Grande, Taylor Swift, Macklemore & Ryan Lewis, and others. Because of this representation, a majority of the lyric alterations thus came from these artists since their music was covered the most by Kidz Bop. Our bubble visualization helped identify Taylor Swift, Ed Sheeran, Drake, Bruno Marks, Katy Perry, Ariana Grande, and Macklemore & Ryan Lewis as some of the artists who account for many lyric alterations.

We also observed that many non-profane words that were associated with more adult friendly behavior were either removed or altered. These words include “drink”, “drunk”, “cigarette”, “smoke”, “wine”. Several religious words such as “God”, “Heaven”, “Jesus”, “sin”, and “demon” were mostly removed from the Kidz Bop version. We also saw common language such as “man” and “woman” altered to “boy” and “girl” to accommodate a younger audience. Overall, we see a very conservative approach to language, and our treemap visualization allows the user to explore all the unique words that were altered or removed and draw their own conclusions.

One area of improvement that our data set could see is filtering out repeated lyric alterations. We noticed that certain songs would have repeated lyrics in the chorus or verse, and our data set would count these repeated lyric alterations as unique data entries. While this only affected a small portion of our data, an improvement to our data set would be to identify unique lyric alterations.

For our treemap visualization, we wanted to focus on this question of “How are words altered in similar or different manners across different artists and songs?” Using a treemap, our size encoding channel helped identify which words in relation to others were altered the most each year.

VIII. Discussion

We maintained the same dataset between our A4 and Final Project and used feedback from both of our milestones to understand how to improve our project. Our goal was to provide a fun, creative, and interactive approach to visualization censorship and lyric alterations of music. Because the data set was imported as a standard CSV, we had the freedom and flexibility to choose how we wanted order and structure our data in a certain hierarchy.

Overwhelmingly, our feedback from the course was positive, and we saw one trending question that the audience wanted to observe: “How do are certain words altered across different songs and music?” This question and general feedback motivated us to use a treemap to show relationships between the censored or altered words.

We also received feedback that our Bubble visualization is an engaging and interactive depth-first visualization tool that allows a user to discover and make their own insights about our project.

Due to the nature of some of the profane nature of work, we also received feedback about implementing a potential profanity filter in our visualization, so our audience is not immediately show expletives.

IX. Conclusions & Future Work

As discussed previously, we saw a strong and conservative approach to the language, and in some cases, the context of the entire song was altered or removed. As a Kidz Bop brand, the data and corresponding visualizations affirms their mission to provide kid-friendly and kid-appropriate music to children.

We are very content with the final product and our two interactive visualizations. Music censorship in public spaces is often something we expect and don’t really question or think past a superficial level. We hope that our case study on Kidz Bop music helps us understand how the rest of the music industry treats lyric censorship and alterations.

Our goal was also to provide a fun and creative visualization tool, and we were happy to enlist two underutilized techniques – treemap and bubble visualizations – to build interactive lyric analysis tools for users to explore and discover unexpected alterations in their favorite songs.

X. About the Authors

Nico Salinas, Shannen Wu, and Jessica Yin are all senior undergraduate students studying Computer Science & Engineering at the Massachusetts Institute of Technology, under the Department of Computer Science and Engineering. This project comes from 6.859 Interactive Data Visualization course from Spring 2021. All three of the team members are former students and experience staff members of MIT’s student-led web development course and competition, 6.148 web.lab.

XI. Acknowledgment

We would like to thank the teaching staff of MIT’s 6.859 Interactive Data Visualization, Spring 2021, and the MIT Department of Electrical Engineering and Computer Science for supporting us in designing, building, and iterating on our project.