## horizontal line



Design Report

31.5.2020

Group Members

Tsz Kin Chau tszkin.chau@student.kuleuven.be

Fengan Li fengan.li@student.kuleuven.be

Hugo Enrique Montaño Castillo hugoenrique.montanocastillo@student.kuleuven.be

Lars Magne Tungland larsmagne.tungland@student.kuleuven.be

Martina Verna martina.verna@student.kuleuven.be

Xin Wang xin.wang@student.kuleuven.be

# Report Goal

The goal of this design report is to justify the design of each visualization, especially with respect to possible alternatives.

# Instructions

Fill in the template provided on the following pages for each of your project's visualizations.

*Note: If a chart on itself is meaningless (you don’t expect the user to perform any task by looking at that chart individually), you can group multiple charts until there is a task a user would be able to fulfill with them.*

[12:07, 5/20/2020] Jim Wang: Final report: 3 pages, explain the effort that went into the project, add reflections, what vizes chosen and why, how work distributed in group, DUE 30 May

Design report: per viz or group of viz, (for us I think it naturally breaks down to each chapter), must specify alternatives, reference course material

[12:08, 5/20/2020] Jim Wang: Design report: 50% of course grade, elaborate benefits/downsides of each choice,

[12:13, 5/20/2020] Jim Wang: Design report: alternative for each viz (viz is defined as for one purpose), Diego/Katrien preference for in depth discussion of comparison for one/two viz and alternatives, no page limit, think about plotting encoding in table, more specific is better

[12:14, 5/20/2020] Jim Wang: Design report: DUE 17 June, 2-3 pages for describing alternative, look at example for guidance on structure and complexity

# Visualization Design Report Template

### Another Viz

## XXXScreencap here

### Chord Diagram - Hot Song Titles

A picture containing ride

Description automatically generated

# Insight

### Another Viz

## XXXWhat Is the Insight

### Chord Diagram - Hot Song Titles

Hot titles use uncommon word pairs to create descriptive and memorable titles. Poorly-ranked songs use more common words. We can also use this to create new titles for existing songs.

|  |  |  |
| --- | --- | --- |
| **Artist** | **Original Title** | **Generated Title** |
| Kesha | Tik tok | Rest the Day, Out All Night |
| TLC | No Scrubs | You Ain’t Got It |
| John Denver | Country Roads | Wake Me Up On the Way |

Some common words are not here, most noticeably being the word “Love.” “Love” is one of the most used, or overused words, in every bottom-ranked song. Great love songs describe the theme, they don’t need to put it directly in the title.

# Data

What data does the user see in this visualization? Only variables that are encoded in some way in the visualization (which can include a tooltip encoding) should be listed here.

### Another Viz

### Chord Diagram - Hot Song Titles

Each node represents a word which is used in a song title. Each link between nodes represents a pair of words which are used together in a song title.

The size of each node on the border represents the relatively popularity of the word. The magnitude of the linkage (occurrences of each word pair in all song titles) is shown as a tooltip.

# Tasks

Why should the user utilize this visualization? What task do you expect a user to fulfill with this visualization? You should list at least the main task, but you can include some secondary tasks that the visualization also allows the user to do.

### The other viz

1. **Main task: Describe, Explore, Create**
2. Secondary task
3. Secondary task
4. Packed Bubble Chart Task: **Create** song lyrics using the words from the Million Songs Dataset
5. Chord Main Task: find new and interesting song titles

### Packed Bubble Chart – Create Your Own Lyrics

1. **Allow and encourage user to create lyrics using the words from the Million Songs Dataset**
2. Explore word connections interactively and uninterruptedly
3. Gain insight about the majority genre, popularity and uniqueness of the connected words.

### Chord Diagram - Hot Song Titles

1. **Allow user to create new and interesting song titles**
2. Highlight differences in word usage between the hottest and least hot songs

# Design Decisions: Visual Encodings & Interactions

How are visual encodings constructed in terms of design choices? What interactions are supported by the visualization system? What other design decisions were taken?

### Packed Bubble Chart – Create Your Own Lyrics

A picture containing screenshot

Description automatically generated

Principles:

The guiding principle is to find a balance between level of details and cognitive burden to a user. Being an exploration environment as well as a lyrics creation application, the learning curve is inevitably steep. New knowledge a user needs to acquire when using the tool includes, parts of speech (POS), song genre, popularity-uniqueness, and the connection metaphor. Thus, a considerable effort has been devoted to simplifying UI/UX to encourage the user to ‘explore and create’.

Component of the visualization:

1. Mega Bubble Chart: Located on the left of the dashboard, it renders all the word appeared in the Million Songs Dataset. User can see the detail of a word bubble via tooltip and choose a word of interest by clicking it.
2. Connection bubble chart: Located on the right of the dashboard, it renders the chosen word and its connected words.

Colors:

Coloring of the nodes represent the major genre of a word. Color scheme follows the convention in the previous visualization.

Background is tuned to medium gray to support nodes with very high or very low brightness value so no nodes will be receded to the background.

Number of nodes:

The control of number of nodes has been one of the most difficult tasks in building the bubble chart. To be an effective tool for lyrics creation, at least a user needs access to noun, verb and adjective / adverb. Without POS division, users have a high probability to get noun connection because the dataset is biased towards noun. Also, in the dynamic of lyric composition, a user needs to pick the right POS to construct verses.

Moreover, we also have to leverage popular words (high term frequency) and unique words (high inverse document frequency). There are frequently appeared words such as ‘love’ and ‘baby’ and unique connections such as ‘forest’ -> ‘enchanted’, ‘spiritual’.

For the first challenge, we have to find a way to separate POS so a user may focus on the designated POS. For the second challenge, we have to find a way to return words from both the popular side and the unique side. As color is reserved for majority genre and size for popularity, virtually there is no further mean of encoding available in the Tableau packed bubble chart.

Initial attempt is to partition the bubble chart into 6 parts, representing popular noun, verb, adjective/adverb and unique noun, verb, adjective/adverb respectively.

A picture containing screenshot

Description automatically generated

This layout has a few issues. First, it is difficult to find a focus among six partitions and this merely increases the cognitive burden of a user. Second, because Tableau will size the bubbles according to the active data in the view, thus, even with lower frequency, unique words have a same scale as the popular words.

Much effort is invested in suppressing the excessive details in the fragmented six partitions layout. By combining top popular connections and top unique connections with top\_n\_filter, and by putting up a dropdown menu for POS option. Finally, a single bubble chart is layout is made possible.

top\_n\_filter is defined as,

IF ATTR([word\_genre\_label\_mask]) = '\_self' THEN "your selected" ELSE

IF [rank\_top] <= 20 - ([connection\_threshold]-1)\*10 THEN "a popular" ELSE

IF [rank\_bottom] <= ([connection\_threshold]-1)\*10 THEN "a unique" ELSE NULL END END END

The final layout contains the node count of 1 chosen word plus 20 connected words.

Size of nodes:

The size of the nodes is proportional to the harmonized word count. In Create Your Own Lyrics, node size is encoded to express the popularity (frequency) of a word which in turn allows the user to learn the popularity difference between the chosen word and the connected words. The challenge here is that the returned 20 connected words are derived from two different ranking, top term frequency and top inverse document frequency. Therefore, a metric common to both ends is necessary to interface the sizing.

A number of metrics are tested for sizing. Namely,

MAX([Count])

AVG([Count])

﻿MAX([termFrequency])

MAX([Count])/AVG([Count])

sqrt(SUM([Count]))

|  |  |  |
| --- | --- | --- |
|  | A close up of a logo  Description automatically generated |  |
| ﻿MAX([termFrequency]) | sqrt(SUM([Count])) | AVG([Count]) |
| Comparison between three metrics. Note that in navy blue the chosen word has 616 occurrences while the connected words have occurrences between 19 and 16,869. | | |

In the end, sqrt(SUM([Count])) is picked as the harmonized word count.

Relation between nodes:

The bubble chart is a representation of a ‘origin-edge’ relation in which the chosen word is the departure point (origin) and the connected words are edges. The relation is realized by always positioning the chosen word (colored in navy blue) at the centre.

|  |  |
| --- | --- |
| page159image61378000 | A close up of a logo  Description automatically generated |

Interaction: Wildcard Filtering

User is allowed to search a word from the mega bubble chart to explore word connection. Wildcard filtering is used in which a user may feel that he/she is progressively navigating in the word bubbles. This is to immerse the user in the meta-narrative of ‘lyric space odyssey’.

|  |  |  |
| --- | --- | --- |
| A picture containing food  Description automatically generated | A close up of a logo  Description automatically generated | A picture containing device  Description automatically generated |
| search string: empty | search string: ‘for’ | search string: ‘forest’ |

Interaction: Exploring Word Connection

Clicking any word bubble in the dashboard will trigger the connection bubble chart to display its word connections. To encourage a user to explore word connections, the interaction is designed to be as fluid as possible. The main issue is that it is easy for a user to lose his/her whereabout after exploring a few rounds: which word am I exploring?

The initial attempt is to use a word bubble dedicated as the indictor of the chosen word. During our internal review, we discovered that it is not intuitive for people to recognize the dedicated bubble as the chosen word. Then, we tried to modify it by adding a scenario sentence ‘connection to <word> in <genre> songs’ to provide contextual information.

A picture containing screenshot

Description automatically generated

This could not solve the problem as well. First, we think that the scenario sentence adds complication to the UI. Second, both the dedicated word bubble and the scenario sentence inevitably cause the user to move their eyeball back and forth. We consider these an increase of cognitive burden and a hindrance to the fluidity of the interaction.

Then, we try to solve it by embedding ‘my whereabout’ within the bubble chart itself.

|  |  |
| --- | --- |
|  | A picture containing stereo, sitting, display, holding  Description automatically generated |

We tried to make use of pre-attentive processing of the brain. First, we added the chosen word in the connection bubble chart and colored it in navy blue. It is based on the fact that our brain can quickly detect object with distinctive coloring. Second, we instructed Tableau to always position the chosen word at the centre. It is based on the cultural phenomenon that centrality tends to have a special status. Third, we made use of the highlighting function. Once a user selects another word to trigger the re-render of the connection bubble chart, the newly chosen word will be highlighted.

We discovered that our brain can quickly learn the paradigm and the connection bubble chart is made self-explanatory. The value of this more fluid version is that it encourages the user to make more connection exploration. This aligns will our meta-narrative of ‘lyric space odyssey’.

Interaction: Panning Popularity-Uniqueness

User is allowed to pan across the Popularity-Uniqueness spectrum with a toggle. Learn may learn the relative popularity or uniqueness of the chosen word by comparing with its connected word.

|  |  |  |  |
| --- | --- | --- | --- |
| A picture containing photo, sitting, table, white  Description automatically generated |  | Full-Popular |  |
|  | Mixed |  |
|  | Full-Unique |  |

Other Interactions: Lyric Composer, Genre Switch

User can use the lyric composer to compose lyric. Action available includes add word, backspace and reset. Lyric composer is an Node.js app embed as a web object in Tableau Dashboard. Word collected in the lyric composer can be retrieved again as a data source of Tableau. Additional analysis of the created lyric is possible though it is not implemented in the scope of this project.

Genre switch allow user to navigate to the word connections of songs from another genre. By default, clicking a word bubble will show the word connections in the reprehensive genre of the chosen word. Again, color scheme of the genre follows the convention in the previous visualization.

|  |  |
| --- | --- |
| A screenshot of a cell phone  Description automatically generated |  |
| Lyric Composer | Genre Switch |

### Chord Diagram - Hot Song Titles

Principles:

The driving principle of this chord diagram is ease of interaction. Depending on the luck and interest of the user, they may or may not find a spectacular new title. However, they should feel as if they could and understand how to do so if they spent more time with it.

Every element starts out presented with equal prominence. Admittedly this is somewhat overwhelming at first. However, as soon as the cursor moves over any element, all other elements are immediately suppressed and the mouseover element remains highlighted. Aided by the dark background, the highlighted chords give the user a pathfinding mechanism enabling exploration. In this way, I help align the presentation of the data to the user’s visual focus.

Colors:

Background: I used the RGB value [50, 50, 50]. A dark background greatly enhances readability of fine, light-colored lines. I found that a pure black background creates slightly more uncomfortable contrast against the brightly color palette.

Text: Wheat (#F5DEB3) gives a pleasant contrast to the dark background and does not replicate any of the colors of the chords. Thus, it avoids creating confusion about labels being associated with more than their individual node.

Node and Edge: For convenience, I chose to use D3 palette [Categorical.Set1]. For a diagram of ~30 nodes, each color is repeated about 3 times. To mitigate the impact for those who are red-green colorblind, the two colors are never placed adjacent to each other. Furthermore, bright edges against dark background would still help with finding connections.

Fonts:

Size: Given the density of data, fonts are sized to maximum that will fit on screen. This ultimately became dependent on maximum word length. Height is the limiting factor for desktop displays since they are typically 16:9 aspect ratio. However, width is the limiting factor for portrait mode mobile displays. For example, the Samsung Galaxy S10 is 3200x1440 and the iPhone 11 is 1792x828; both are 9:20 aspect ratio. Theoretically, you can encourage users to turn their phones but it’s an extra second of work which is a meaningful amount. Alas, the old 4:3 aspect ratio is best for circular diagrams but CRTs are as extinct as floppy disks.

Font selected: Arial is the classic and well-suited for our task. It’s web-safe, free and readable. Looking back on it, the kerning is a bit smaller than ideal. Given the odd angles in which text appears, more generous spacing would help with readability even at the expense of screen real estate.

Rotation:

This was a difficult decision. Horizontal text labels improved label legibility but diminished the relationship between labels and nodes. Rotating text labels improved the natural flow of the eye from label to node to chord. However, labels at the very top and bottom of the diagram are difficult to read. Ultimately, I chose to prioritize maintaining the circular theme of the chart. To aid the user, all tooltips are displayed horizontally.

Layout (in the Tableau Story):

Since chord diagrams are not as well-known or self-explanatory, I chose to start by giving the user some samples of what they expect to find. Even if the user does not explore the graph themselves, they don’t leave the slide empty-handed.

Nonetheless, I hope that the large brightly colored diagram in the middle of the page would encourage users to play with it for at least a bit. Please note that the for the sake of time and consistency during the presentation, we replaced the live chart with an animated gif showing how to create the title “Rest All Day, Out All Night.”

Desktop/Mobile:

Because of the number of nodes, this diagram is best suited for desktop usage. While it does dynamically scale, zooming out becomes problematic as discussed in below in “Nodes and Edges”. Mobile also has no concept of cursor which makes the on hover effect much less useful. The alternative using Holoviews is far better suited to mobile use.

Number of nodes and edges:

This was a huge part of testing and probably the most difficult design decision. The more nodes and edges, the more complex and interesting connections are revealed. However, there is a massive tradeoff in usability and readability beyond a certain point. A chart with fewer than 20 nodes dies a rather dull death; there are simply not many interesting titles.

Chaos springs from charts with more than 100 nodes. Nodes are difficult to pick and the thin lines are difficult to follow. Each edge node has an average angular size of 25 pixels. In practice, less popular words will have an edge length of <10 pixels. Compare this the average cell height in Excel of ~40 pixels – I would be limiting the usability of the chart to Excel wizards and competitive FPS gamers.

Size of nodes and edges:

The size of each node is proportional to its term frequency. This creates a problem where common words (e.g. “the”, “you”) take up a huge amount of screen space, thus squeezing other words. Less common, but more interesting, words then become difficult to select. To resolve this, I manually adjusted the raw frequency using roughly a log scale. What this detracts from scientific accuracy, it enhances in usability.

I also implemented a cut-off so that words with fewer than 4 total connections are excluded from the final output. These terminal words add little to the variety that a user may find.

References:

My implementation and customization use the D3 wrapper by https://shahinrostami.com/ and package from http://holoviews.org/.

# Alternatives

*Note: you should include at least 1 alternative. If you haven’t thought of any alternatives during the design process, think now if you could not use the visualization described before, how would you visualize the same data to support the main task.*

Describe as many alternatives as you wish and justify why your encodings or the design of your visualization may be better. You should use, when possible, research results or heuristics from class material or bibliography. If you are not able to do that, explain your reasoning for taking the design decision. Note that the data used for the alternatives and the task that they try to fulfill should be the same.

## Alternative Visualization 1

Describe the alternative chart or add a screenshot of it. If you have not implemented the alternative chart you can add some sketches, images from a similar concept with a different dataset, or any other graphical support that helps understand the alternative chart.

### Design Decisions

Describe how you have or will encode the previously described data in the alternative chart.

### Comparison

Justify why the design of your final visualization is more effective for the described tasks. Note that the final visualization should be more effective on the listed main task, but an alternative visualization can be more effective for some of the secondary tasks.

*Note: to get a better idea of how these justifications can look like, you can read through this Lisa Charlotte Rost post:* [*https://blog.datawrapper.de/dualaxis/*](https://blog.datawrapper.de/dualaxis/)*. In the article, she justifies how dual charts are a worse alternative than, for example, having two line charts side-by-side, not only by explaining intuitively how dual axis can be misleading but also by research results. In particular, she uses results from the following article:*

*Isenberg, Petra, et al. "A study on dual-scale data charts." IEEE Transactions on Visualization and Computer Graphics 17.12 (2011): 2469-2478.*

## Alternative Visualization 2 - Holoviews for Chords

### Design Decisions

One alternative I explored thoroughly was to use the Holoviews framework for creating a similar chord diagram. The encodings would be similar. Nodes would represent word frequencies, edges would represent the bigram occurrences of each pair, and the number of edges would represent

The major difference would be in functionality discussed in detail below. Holoviews is less dependent on cursor movement and placement.

In summary, Holoviews is a far superior solution for mobile and for balanced datasets.

One alternative which I considered was to remove stopwords entirely. It would mostly solve the problem outsized nodes. However, many stopwords are very useful in creating grammatically correct titles. In a user’s travel around the diagram, they are simply too useful as origins and intermediate stops that I chose to leave them.

### A picture containing ride, fireworks, device Description automatically generated

### Comparison

|  |  |  |
| --- | --- | --- |
|  | **Holoviews** | **Chord** |
| **Zoom** | Built-in support for mousewheel or pinch-to-zoom | Browser zoom only |
| **Pan** | Drag to pan | None |
| **Dynamic Scaling** | Yes | Yes |
| **Tap / Hover** | Clickable nodes highlight and lock the selection. Hover over tooltip. | Node highlighting on hover only. Hover over tooltip. |
| **Reset** | One click to reset to default state | Automatically reset when cursor moved off-screen |