ARTEMIS

PROCESS BOOK

COM-480 DATA VISUALIZATION

May 28th 2020

GROUP

Artemis

TEACHER ASSISTANT

Jean-Baptiste Cordonnier

TEAM

Franck Dessimoz - 246602

Simon Roquette - 246540

Justine Weber - 261458

TIMETABLE

The origin

The starting point of this project was the dataset : *"50 years of pop-song music"*. It contains the lyrics of the yearly top 100 billboard songs from 1965 to 2015. The three of us are into music, and we wanted to combine NLP and data visualization. Based on our research and the demonstrations from the course, we were particularly interested in visualizations including a time or a space dimensionality.
We thought it would be very interesting to create a visualization showing the influence of historical events on pop culture, and that's where it all began.

For that purpose we had to gather events data. Since we could not find any relevant dataset for our project, we created our own by scrapping information from https://www.onthisday.com. This site has the advantage of showing only major events which would perfectly fit our needs. Indeed, for each year from 1965 to 2015, we were able to extract the day, month and year of the event, as well as a one-sentence summary. Having collected these data, we used the Wikipedia Python API, to gather the article's url associated with the event as well as the summary of the article. At the end we were able to gather 1115 events over 50 years.

A challenging journey

The first challenge we faced has been about the data. To be able to implement our idea, we had to have two datasets : one about events and one about data. This requires a lot of processing and scrapping. We knew it was not the objective of the course, but we still decided to spend quite some time on this, in order to get something interesting, and because we needed to have precise data for the NLP.

Once this was done, we were only at the beginning, because the biggest challenge regarding the data was to link events with songs, through NLP techniques. This was ambitious, and our solution is far from behind perfect. As NLP was not the goal of the project, we could not spend too much time to try and improve model performance. In the end the results are not always very satisfying. Anyhow this idea was experimental and is meant to be further explored. The NLP matching process we have finally implemented is described in the following section.

Now about the core topic - animations - we experienced some twists and turns.
Once we had all the static components, one first animation we tried to implement was zooming on the timeline.
Since none of us had a previous experience using d3.js, it was very hard to design all the elements right the first time. We had to change a lot of things along the way.
At first we built a static timeline, and tried zooming with an svg component. In the end this solution was very complicated to implement, and upon advice of our TA, we went for another solution, using time axis.

This changed a consequent part of the code : we then defined the points' position by their x coordinate on the timeline.

Once all the animations were done, we experienced some issues with the website's smoothness. We spent some time optimizing the website and trying to understand what may cause this lack of smoothness.

At the end, we had to fix a huge amount of small details, which don't seem like much, but which actually require a lot of effort, time, and patience.
One example was making the dates move to the top of the timeline when we open a song description, and making them move down when one closes it.

To be honest, understanding d3.js's logic has not been easy. It was sometimes very confusing to think the way d3.js works, especially for accessing data information.

Finally, because of the current crisis we were not able to work all together in the same room, and this has been a real obstacle for some parts, such as the brainstorming, or the coordination when coding, especially for designing a common stable architecture for the code.

We could tell you a lot more about those challenges, but we thing you get a good overview of the types of challenges we faced through these examples.

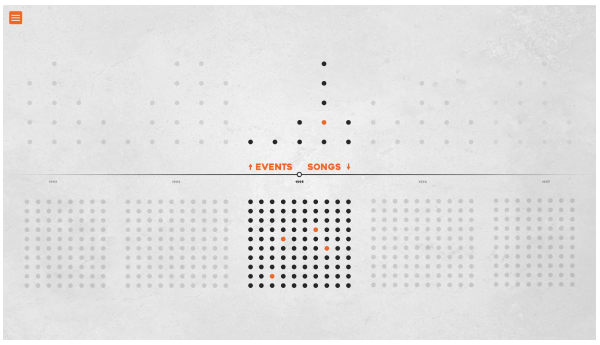
NLP matching process

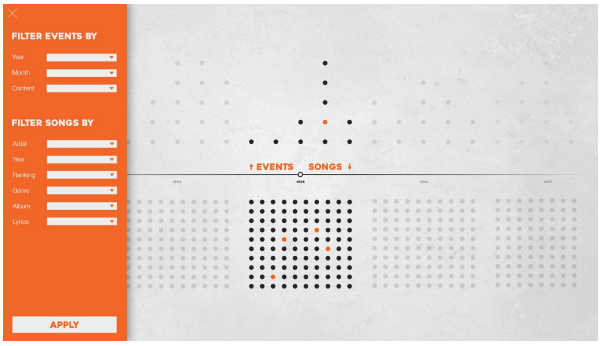
To find references between song lyrics and events, we tried a various strategies. Most unsupervised classifying techniques in language use semantic to determine proximity between sentences. They often have thresholds regarding the number of words in the texts to classify, and also perform poorly when classifying texts of unbalanced length, which is often the case for us (a short event description vs a 1000 words song…) .
This is why we decided to use NER (Named Entity Recognition), which allows us to extract Entities such as Person, Date, Event, Political / Sociological group… An event and a song are then considered to refer to each other if they have at least one entity in common. We used a "soft comparison", by saying they one of the entities must contain the other. For example, if a song refers to "Obama", and an event to "President Obama", they will considered to be matching. The NER we use comes from Spacy, and, even though it is trained on english, it also works reasonably on other languages, and we happen to have some lyrics that are not in English so it is useful.
By just doing this, we had ~60 references per song, which seemed like a lot, so we further applied a text classification method based on a pretrained model of ALBERT (light version of BERT). We vectorized both lyrics and events (with the first 512 words, all further are ignored) and then computed cosine similarity between the vectors. We believe this might not be the best tool for this particular case, with no fine tuning of ALBERT considering our results. Indeed, we saw that each song lyric had on average a cosine similarity of 0.74 to the events... We could have switched to Tf-Idf vectorisation or Doc2vec, which we both had implemented in our notebook, but we decided to focus more on the data visualization, as this had already been very time consuming.

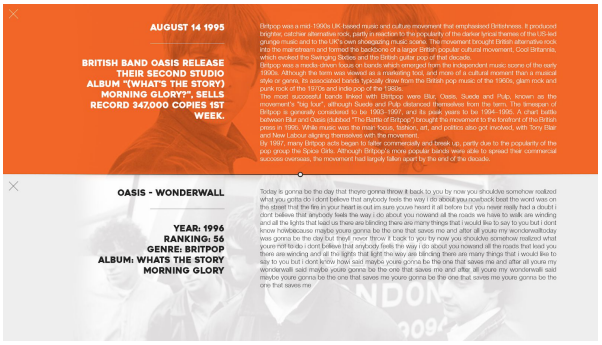
The final decision process to consider if a song is making a reference to an event is :
If they have at least one entity in common, and the cosine similarity of their vector is > **0.92**, then it is a reference.

Evolution of our visualization

For the first milestone, we created the following sketches:







One will see that our implementation is a little bit different. Although the structure and global design are the same, the points arrangement is now more disorganised.

The x positions of the event dots, is based on their date. We have not related an event to a specific year as time is continuous.
The y position of the event points do not have any specific meaning. As there are quite a lot of events, we decided to spread them in the space. Having them all on a line would have made it harder to distinguish two points. Also having a big empty space didn't not make much sense and we decided to make the points take all the available space above the timeline.

For the background of the descriptions, it was quite a challenge to find images for each song / event and apply an orange / grey filter on them. Thus we decided to focus more on the animations. This would be an interesting improvement to our visualization though.

We thought it was not necessary for a user to select a specific year, but rather to zoom on a period of time (including several years). Moreover, a song can be linked to an event which has appeared a few year ago. This is why we did not put this white dot and the grey / black contrast of years which are selected or not.

Main design decisions

Enumerating all the design decisions we took would be very tedious, so we will only focus on one, which took us some real thinking.

The question arose as to how to let the user navigate through our visualization. Two behaviours were envisaged. Here we describe the scenario where we start from the events to discover songs, but the explanation can be transposed symmetrically.

1. The visitor clicks on an event.
2. Then he choses a song (linked to that event somehow).
3. Two options
    - **Wandering >** He can then close the event description and navigate by keeping the song as a new reference.
    - **Guiding >** He cannot close the event description, because it is his reference for the navigation. We force hime to close the song description if he wants to go any further in the website.

We chose the guiding option. Although the "wandering" option had its advantages, by allowing the visitor to navigated indefinitely, we thought it may be too confusing and he might get lost in time.

Peer assessment

For the first step of the project, the work was separated as such :

- Franck took care of the data preprocessing and cleaning, as well as enriching it through scraping. When we decided to include the events, he took care of retrieving information from Wikipedia.
- Simon focused a lot on the NLP part, analysing the lyrics and the events descriptions, and founding inventive ways to relate both through several NLP methods.
- Justine worked on the inspiration part : browsing the internet to find interesting visualization and analysis with our dataset.

For the visualization idea, we had a big brainstorming session, where each of us came with a proposition and we decided together what ideas to pick from each proposition. The three of us participated the decision process, to end up with the following proposition : a unique full-screen visualization, with a central timeline and dots representing the events and songs, up and down.
The design proposals were made by Franck. The color palette as well as the font and the background were his ideas and Simon and Justine went along with it, as it looked very nice.

For the website development :
Franck started with the backbone of the website. He built all the static components, and was in charge of the styling (css). Simon took care of creating the server with flask, and later to host the website on github.io.
Then each of us worked on some part of the animations. Justine focused on the timeline interactivity and the zooming/unzooming functionality, adapting it specifically . Franck focused more on handling the data, by representing and animating the dots for each data point, and navigating through the website (opening and closing description for example). Finally Simon took care of linking the events with the songs visually, and highlighting how the matching was done with the entities. The details for each of these tasks is described above. For the rest of the animations and the details, the workload was split between the three of us, depending on how fast we were moving on the major tasks.

Further improvements

We feel very satisfied with the rendering, and are proud to have been able to complete this project.

Nevertheless, it has been hard to finish it, because we constantly wanted to improve it, bringing in new ideas. There are still a lot of ideas we have and we will briefly mention here.

- Having a demo at the beginning, show a new visitor some interesting links he may find.
- Specifically highlighting the matching entities when displaying a song and an event description at the same time. We started this part, but there were too many limit cases to handle, so we decided to leave it as a future improvement.
- Showing the matching percentage.

- Improve the NLP model to get more relevant matchings.
- Enable people to choose the matching percentage threshold they want.
- ...

Conclusion

Through this project we learned a lot, and we are very proud about the result.
It was a really exciting project to work on, and we hope we managed to share this thrill with you !

Special thank to J.-B. Cordonnier who was available and really helped us when we were stuck with some problem.

Have fun and get lost in time !

Justine, Franck and Simon