Social Compassion — A #PrayForSyria Visualization

Carlos E. Marciano[a]

[a]*Federal University of Rio de Janeiro, Brazil, December 2018*

Visualization: *https://cemarciano.github.io/Social-Compassion/*
Source code: *https://github.com/cemarciano/Social-Compassion/*

Keywords: social data analysis, *Twitter*, Syrian civil war

Abstract: Despite the ongoing Syrian civil war being a monumental demonstration of cruelty and indifference, the *Twitter* trending topic *"#PrayForSyria"* has been widely used over the years to express affection and compassion towards those who suffer the most. In this context, this study collects 222,180 *tweets* from March 2011 (when the conflict started) to December 2018, showing how the *Twitter* network came together to discuss and spread the word about the Syrian conflicts. The resulting visualization is presented in the form of "word clouds", where the size of each word is proportional to its frequency. Clicking on a word reveals 20 random tweets which contain that specific word, while a dynamically recorded instrumental track contributes to an overall atmosphere of sorrow and respect.

1. Introduction

On January 26th, 2011, a series of popular protests would begin in Syria. Influenced by other civic demands in the Middle East, a bloody armed revolt broke out with the intent to overthrow President Bashar al-Assad and establish a democratic government. The retaliation, unfortunately, came in the form of chemical weapons and even more deaths. Until 2016, the conflict had resulted in more than 470,000 deaths; 1.9 million wounded; 7.6 million internally displaced people; and 5.6 million refugees, according the Syrian Centre for Policy Research [1].

In an age where online connectivity is the norm, the Syrian civil war provoked a variety of reactions all over the globe. By making use of the trending topic *#PrayForSyria*, a plethora of users across the *Twitter* social network demonstrated their feelings to the world over the past 7 years of conflict. In this study, we aim to collect random *tweets* related to the ongoing war and display them in an interactive manner, while the viewer is constantly reminded of the sorrow and the pain caused by the war.

In general, this visualization is comprised of "word clouds" containing the words that appear the most in all *tweets* related to the Syrian conflict. When clicking on a word, the entire visualization changes to a different atmosphere, while *tweets* containing the chosen word are displayed to the viewer alongside a new word cloud. The resulting work is an interactive experience involving *social data analysis* and *web visualization*, where the viewer molds the content that he or she would like to explore.

2. Data Preparation

In order to obtain a large amount of *tweets* encompassing almost 7 years of social interactions, we developed a framework in the **Python** programming language (file *fetch.py* in the root directory). Unfortunately, the *Twitter API* limits data collection to the previous 7 days, which theoretically would make this project unfeasible. However, other open-source alternatives exist, lifting most of these restrictions and allowing even more scraping functionalities. One of the most robust of these tools is **TwitterScrapper** [5], providing a more granular control of which tweets we would like to obtain.

Moreover, user input is only a fraction of each *tweet* object. In fact, most of the information contained within a *tweet* is metadata, which can range from user location to *retweet* information [2]. As a result, a simple database program is necessary to accumulate the large volume of incoming data, and we chose **MongoDB** for this task. This *NoSQL* database stores data in the *JSON* format, being a highly scalable solution for keeping *tweet* information. After filtering most *JSON* fields from the raw *tweets*, the structure stored in *MongoDB* can be seen in Table 1.

Field	Example
id	957371457184571392
text	Good morning...
user	tetsuya_kato
fullname	加藤哲也
lang	en
replies	0
retweets	0
likes	0
timestamp	2018-01-27 21:55:17

Table 1: Example of *tweet JSON* structure sampled from the *MongoDB* database.

Having obtained all 222, 180 *tweets*, it was then necessary to filter unwanted data. For instance, many *tweets* contained lengthy *URLs* that could detract the viewer from the visualization experience. Moreover, a significant number of *tweets* contained line breaks, which would make them vertically large upon being displayed. Therefore, the *Python* library **Pymongo** was employed to retrieve *tweets* from the database and alter their structure through the use of regular expressions (*regex*). For instance, by searching for whole words containing the *substring* "http", we were able to identify links and remove them from the text. Ultimately, *Pymongo* was also important in obtaining the list of most frequent words appearing across all *tweets*.

In order to select which words would inform the word clouds, we chose the most frequent 300 terms across all tweets. Afterwards, the selection process was non-automatic: entries such as *"the"*, *"as"* and *"I"* were manually discarded, while keeping all nouns such as *"people"*, *"innocent"*, or *"Allah"*, resulting in 247 total words. We tried to avoid introducing any kind of significant bias, and this includes not discarding words based on their language. As a result, although the visualization is mostly limited to the Latin alphabet (since *#PrayForSyria* itself is written in this alphabet), many words deriving from languages other than English were kept, such as Spanish and Portuguese. Some rare *tweets* written in different alphabets may appear, but only because they have also included *#PrayForSyria* and other Latin words alongside their message.

3. Data Visualization

3.1. Visualization Tools

Having prepared all the data to be displayed, we needed to investigate a visualization strategy. By choosing the *Web* as our primary development asset, we were able to reach a wider audience and utilize preexisting libraries that excel at drawing and organizing elements. Particularly, in order to build our word clouds, we made use of the **eCharts WordCloud** library [3] due to its native support for advanced image masking, allowing us to shape the word clouds in any way we wanted.

When choosing the image masks for defining the shapes of these word clouds, we resorted to existing online images and converted them to a black and white format. The shapes chosen for this visualization are supposed to provoke the viewer with war-related themes, including (in this order): a machine gun; a heart-shaped map of Syria; a heart; a bomber aircraft; a family; and a stop sign. The color of each word cloud is randomly generated by using the **HSL color system**: we keep *Hue* at 0 (so that the result is red) and *Saturation* at 100% (so that we get a full color), while *Lightness* varies randomly from 40% to 80% (file *index.js* in the */Visualization/js* folder).

The techniques employed to create the animated backgrounds of each word cloud vary significantly, and can range from rain effects to cloudy skies. Some of these backgrounds are dynamically drawn on an *HTML canvas* through **JavaScript**, while others are simply an animated *GIF*. All of these backgrounds were curated from external online sources, and we only mildly altered some of their aspects (*e.g.* their color) so as to best fit the visualization atmosphere. In order to dynamically switch between backgrounds and word clouds whenever the viewer clicks on a word, we employed the **JQuery** library for *DOM* manipulation, which provides granular fade control for each element on the screen.

Lastly, in an attempt to further immerse the viewer, we recorded different guitar tracks to be played during each word cloud. All 6 tracks share the same chord progression inspired by *Pink Floyd's Comfortably Numb*, which fits the overall atmosphere of the visualization. In order to seamlessly switch between each guitar track whenever the viewer clicks on a word, we utilized the **HowlerJS** library [4], which attenuates most of the issues faced when dealing directly with *HTML 5 Audio* (*e.g.* music loops not playing seamlessly). This combination of visual and audio elements contributes to a more immersive work, where the user is capable of establishing a more personal connection to the visualization.

3.2. Visualization Paradigms

Under construction.

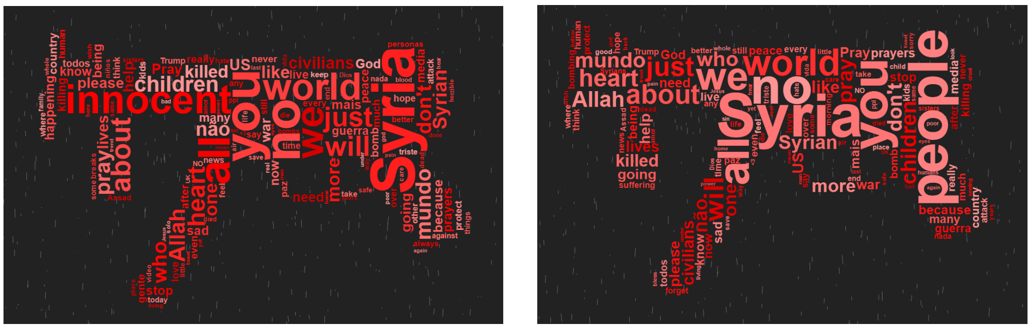

Figure 1: Two examples of the same word cloud with randomized word positioning and sampling. Word sizes are kept the same (*e.g.* notice the word *"you"*).

References

[1] Ian Black. 2016. Report on Syria conflict finds 11.5% of population killed or injured. (2016). Retrieved December 22, 2019 from https://www.theguardian.com/world/2016/feb/11/report-on-syria-conflict-finds-115-of-population-killed-or-injured

[2] Twitter Inc. 2019. Tweet object - Twitter Developers. (2019). Retrieved December 22, 2019 from https://developer.twitter.com/en/docs/tweets/data-dictionary/overview/tweet-object

[3] Yi Shen et al. 2016. ECharts wordcloud extension based on wordcloud2.js - Github. (2016). Retrieved December 21, 2019 from https://github.com/ecomfe/echarts-wordcloud

[4] James Simpson et al. 2013. Howler.js - JavaScript audio library for the modern Web. (2013). Retrieved February 22, 2019 from https://howlerjs.com/

[5] Ahmet Taspinar et al. 2018. Twitter Scraper - Github. (2018). Retrieved December 21, 2019 from https://github.com/taspinar/twitterscraper/