

Final Project Report

Ameen, Chris, Hongxi (Jerry), June

Final Project Report	0
The Artistic	1
The Technical	1
Schematics	3
Where to Go Next? What to do Better?	4
Source Code	4

The Artistic

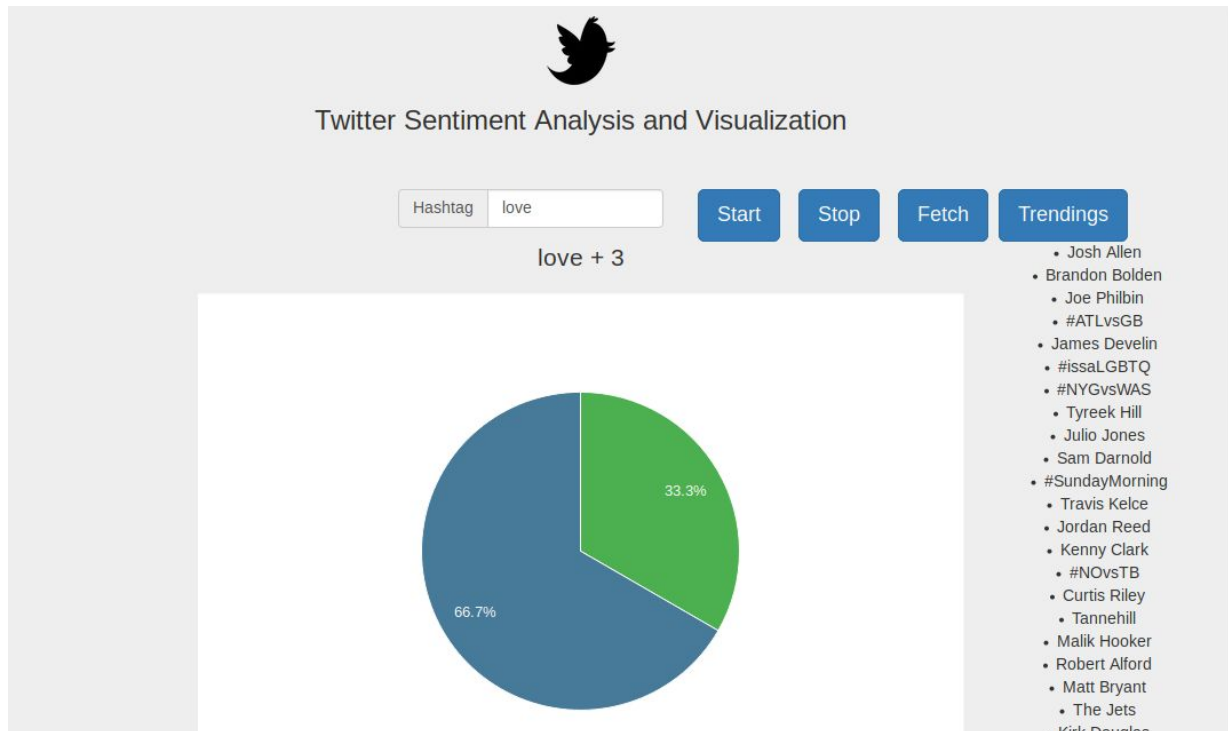
The web is a vast sea of information, yet as users we only ever access it via limited interfaces. The aim of this project was to demonstrate the beauty that the internet is: its immensity, and its role in connecting humanity. Of course, representing the whole of the internet would be a bit difficult... We needed to figure out how we would communicate these large ideas but from a more limited dataset.

We decided we would use Twitter's API. We were to fetch tweets, and analyze them for sentiment. Next were the physical aspects of the project. We decided that our piece should be an art installation, so as to affect viewers on an emotional level; we thought the only way to truly communicate the large ideas we were aiming to communicate was through metaphor and abstraction. Light, colour and sound then seemed natural choices for representation of emotion. A physical aspect was added later: two towers, or figures that would rise and fall in face of each other: a dance. All this was to be presented on a plain white backdrop. This way we would minimize distracting clutter. Thus, the viewer would focus on the shifting colours, figures, and sound, and what these might represent.

The Technical

The web application was made through a framework called django. Django is python framework which eases the implementation of developing web applications. The user interface provides many functionalities to the user. Users can put the hashtags they want to stream within the textbox and then press the start button. The 'fetch' button will update the pie chart and the 'trending' button will search for the 100 most trending topics at the moment and then display them within the screen. A screenshot of the UI can be seen in the following figure.

When a user clicks to start streaming the hashtag they submit, a call is made to Twitter's streaming API. This is done very easily through the use of a library called tweepy. This library will initially start a stream on the hashtags you provide it. When a new tweet comes in which includes this hashtag, a method within python is fired. Inside of the fired method, the tweet is put into another library called textBlob which analyzes the text and gives it a value between -1 and 1. Negative one means that the tweet appears to be very negative and positive one means that the tweet appears to be very positive. This value is mapped to value between 0 – 100 and this value is send through the serial to arduino.



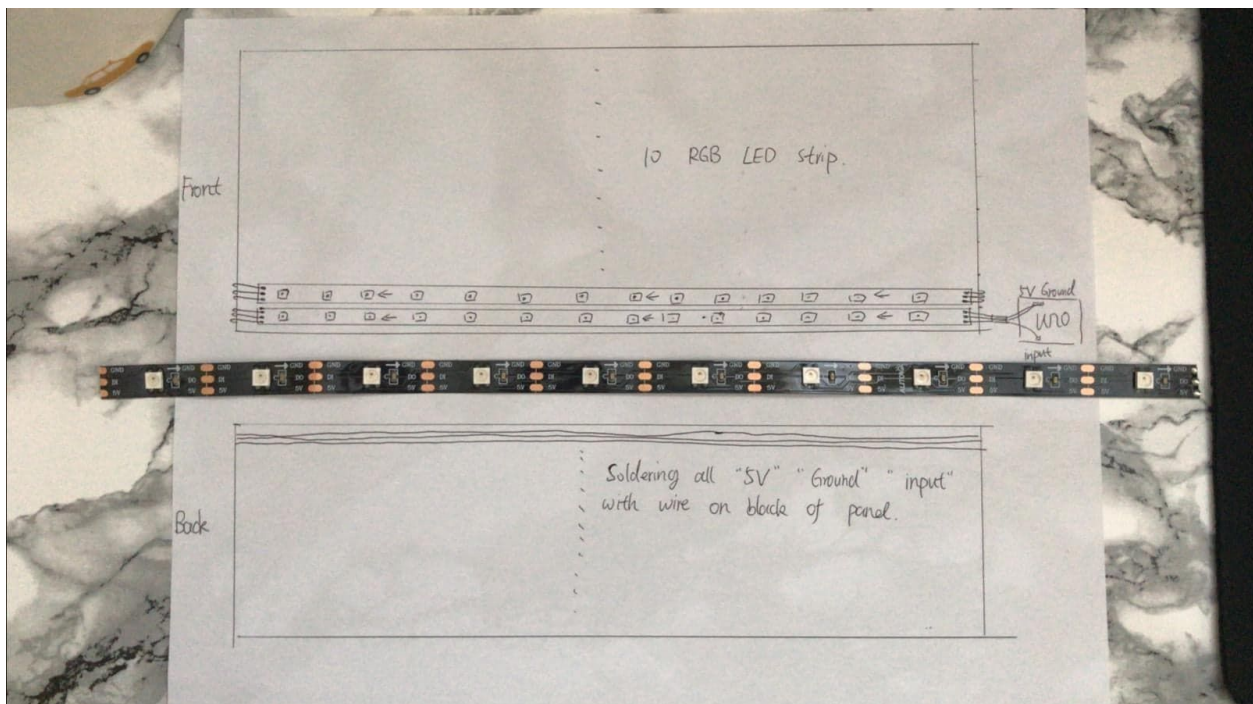
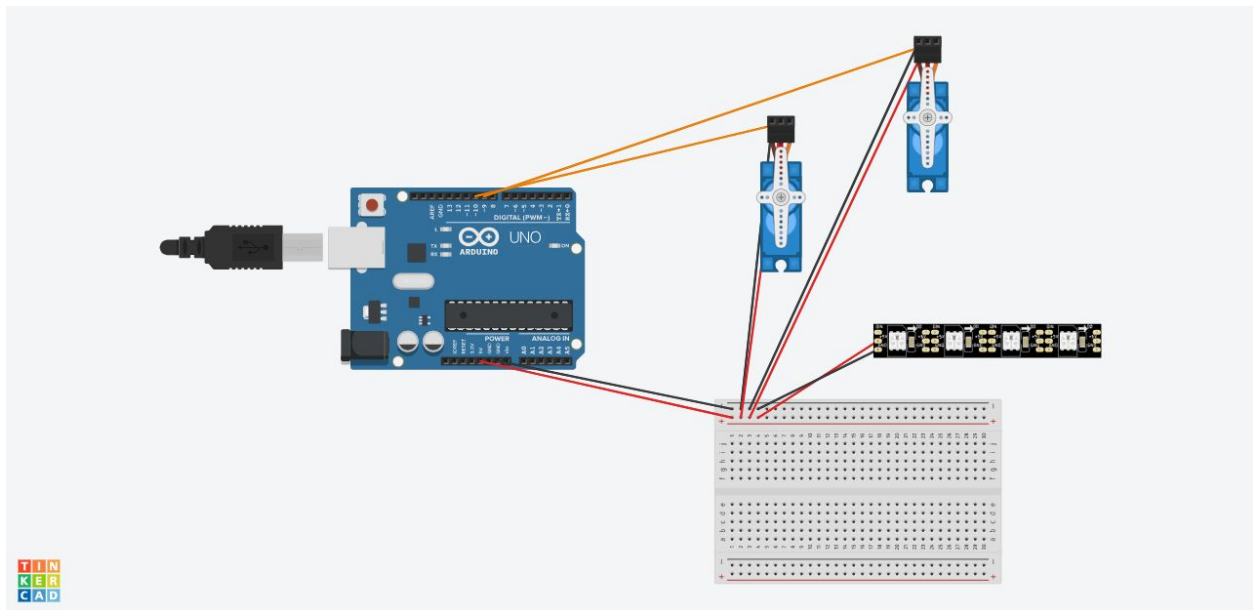
The arduino keeps checking if anything is available through the serial. If there is something, this value is put into a buffer array. From the buffer array, the percentage of positive, negative and neutral emotions is calculated and these values are shown within the screen. We also calculate the average sentiment value of the tweets in the buffer and print that into the serial. This value is used by Max to produce the sounds.

The light display was created using strips of LEDs. These lights were programmed to display a colour ratio based on the average sentiment values of analyzed tweets. Additionally, an animation would play with each new tweet analyzed.

The towers would rise and fall via a pulley/servo system hidden in a chassis underneath the installation.

The sound was created through Max. There are several different oscillators whose properties change according to the values being received from the arduino serial monitor.

Schematics



Where to Go Next? What to do Better?

In the web application, a feature could be added which would display the analyzed tweets along with their scores. Within the physical part, a screen could be added to showcase the processed tweets along with their scores.

As mentioned, this installation would work better if it were upscaled. Additionally, it might have been wise to affect the servos via a different data stream than we did, as their motion was barely noticeable.

Source Code

The source code of the web application can be found in the following git repository
<https://github.com/ameenjaj/TwitterSentimentAnalysis?fbclid=IwAR1jBesPLFiCOufjMWkHuWbUmAZkO72YuHgskb8kMZb9vLVT9zbr-ZfH94U>