WaCC: Trump-sentimeter

Joey Antonisse (s3245543)
Siebert Elhorst (s3264254)
Amey Bhole (s3411427)

November 3, 2017

1 Project summary

The main aim of this project is to create a scalable website which will provide sentiment analysis to Donald Trumps tweets called Trumpsentimeter. The website provides the favourites, retweet count, sentiment which is either positive, negative or neutral and sentiment score for Donald Trumps individual tweets as well as graphs showing an overview of for his latest 100 tweets. This project was created using PHP, Angular2 and Python as the main programming languages.

2 Docker

The server-nodes are simulated with docker containers. Every docker container is a server node with its own operating software (usually Ubuntu or Alpine). Each docker container has it's own hostname which is given via the docker-compose yml (configuration) file. With the hostnames it is not needed to have IP addresses of every container, but a simple name will suffice.

3 The Architecture

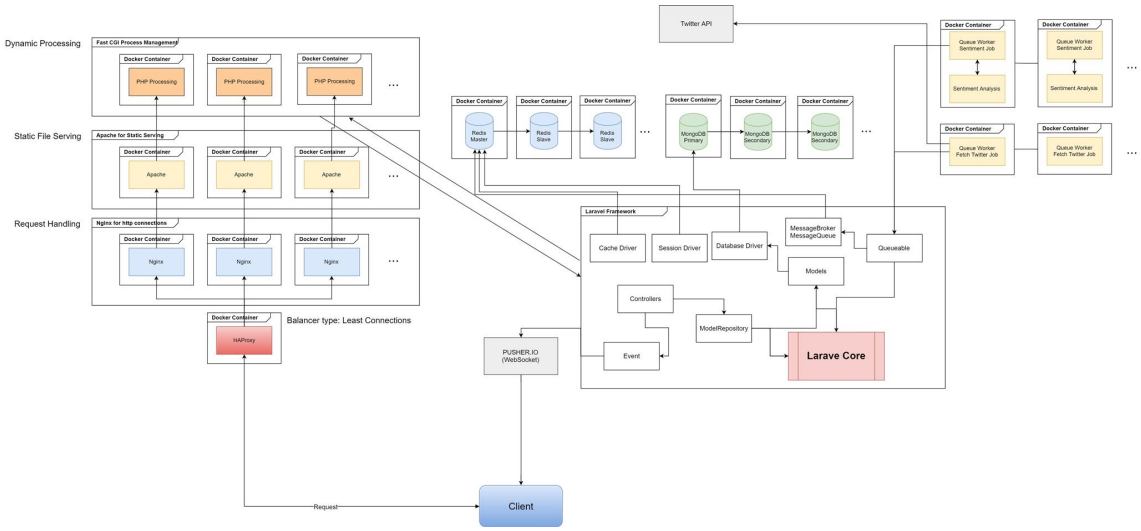

Figure 1: Website Architecture

Figure 1 represents the final architecture of the Trumpsentimeter which consist of following main components:

1. **Load Balancer**
2. **Request handling and PHP Processing**

3. **Laravel framework**
4. **Mongo and Redis databases**
5. **Pusher**
6. **Sentiment analysis**
7. **Queues and Job**

All these components run on a server for 24/7 availability of the website where the sub-components of load balancer, Databases and Sentiment analysis have their own docker container for scalability.

3.1 Load Balancer

This component consists of the following sub-components:

HAProxy (High Availability Proxy): It is an open-source load-balancer which can load balance any TCP service. HAProxy is a free, very fast and reliable solution that offers load-balancing, high-availability, and proxying for TCP and HTTP-based applications.

3.2 Request handling PHP Processing

Ngnix: It is a very efficient HTTP server used to accept the incoming connection. It roves to me much more stable than apache for connections and uses less memory. Also for high request amount the nginx will not have the troubles that apache has with the squared request times.

Apache: Is used to return static files which are rquested by the client. Apache is chosen as a static file serving process because it is proven to be better in file serving than nginx.

Fast CGI Process Management: It is a PHP process manager which carries out the processing of each request made by the client. The codebase of which the request leads to is the Laravel core.

3.3 Laravel Framework

It is the intermediate (code based) system which manages the logic and styling of the web application. it routes the http requests from the FPM and gives back its response. It is a highly configurable framework with a lot of best-practices on the side of application structure/architecture.

3.4 Mongo and Redis database

MongoDB: It is a free and open-source cross-platform document-oriented database program which is classified as a NoSQL database program. MongoDB uses JSON-like documents with schemas. MongoDB is used to store the tweets and the sentiments for it. The database has a primary and a secondary, where the secondary can be scaled infinitely.

RedisDB: It is used to store the data for the jobs from the message queue and broker from the the laravel framework. Like the MongoDB the Redis database is also scalable, it has a master and a slave, where the slave is scalable.

3.5 Pusher

Pusher websocket: They allow a long-held single TCP socket connection to be established between the client and server which allows for bi-directional, full duplex, messages to be instantly

distributed with little overhead resulting in a very low latency connection.

3.6 Sentiment Analysis

The sentiments analysis were carried out by collecting data through a twitter API and the analysis was implemented using a python script. The following libraries were used in the Python script:

Tweetpy: This library is used for accessing the twitter API

Json: It is used to parse JSON from strings or files into a Python dictionary or list. It can also convert Python dictionaries or lists into JSON strings.

Pandas: It is used for data manipulation and analysis of the tweets.

Numpy: It is used for adding support for large, multi-dimensional arrays and matrices for storing and processing of data from the twitter API, along with a large collection of high-level mathematical functions to operate on these arrays.

Textblob: It is used for processing textual data and provides a simple API for natural language processing (NLP) tasks such as part-of-speech tagging, noun phrase extraction, sentiment analysis, classification, translation, and more.

re: Regular expression is a special sequence of characters that helps you match or find other strings or sets of strings, using a specialized syntax held in a pattern. This library was used to pre-process and clean the text received from the tweets for sentiment analysis.

3.7 Queues and Jobs

To keep the website up-to-date with tweets and sentiment analysis, we used queues and jobs as a base for keeping track of what the system has to do. Queues has also the upside that different queues can be split and scaled according to what the process is. Every 2 minutes a cronjob is fired and the whole system will get new tweets. For every tweet the process gets from the twitter API, there is a new job created on the 'sentiment' queue, which will process the tweet for sentiment. These queues are independently scalable, as of now there is the 'default' queue and the 'sentiment' queue. If for some reason 'default' has more load, it can easily be scaled with a new node or docker container.

4 Technologies Used

Front-end: Angular 2, Material Pro, JSON, HTML, CSS, Javascript, PHP, Websockets, HTTP

Back-end: Laravel PHP, JSON, Python for sentiment analysis

Databases: 2 x Mongodb (Primary and Secondary), Redis (Master and Slave)

Other: Nginix, Node JS, Docker, php-fpm 7.0, Websocket, apache, Supervisor

5 Front-end

$Angular2, MaterialPro, JSON, HTML, CSS, Javascript, PHP, Websockets, HTTP$

The front-end is separated in three sections, the homepage which contains the latest 100 tweets of trump, the general analysis of the latest 100 tweets and an analysis by id page which contains specific information about a certain tweet.

5.1 MVC

For the front-end we tried to achieve a MVC model, usually angular 2 is divided into a view and a component. Where component is the model and controller + directives and services. Since we prefer a MVC model we used services as model, component / directives as controller and view as view. In our program we have three services: WebsocketService, HttpService and a DataService (these are the models).

WebsocketService:

The WebsocketService is responsible for the websocket connection, it fetches data whenever new data is sent and puts it in the DataService.

HttpService:

The HttpService is responsible for the http requests, as soon as the page loads it will fetch data from the server and put it in the DataService.

DataService:

The DataService is the most important service, it keeps track of all the data that is fetched from the websocketservice and httpservice. By putting all the data in one service all the data is loaded once and available in all components. All the data is fetched asynchronous, meaning the data is loaded while the user can navigate through the site.

5.2 Homepage

The homepage contains the latest 100 tweets that trump tweeted. It's live updated using websockets, whenever trump tweets the tweet will appear (little delay, because of sentimental analysis) without refreshing the page. Each tweet has a show sentimental analysis option which gives user the opportunity to see some specific statistics of the tweet. In figure 2 the homepage is shown.

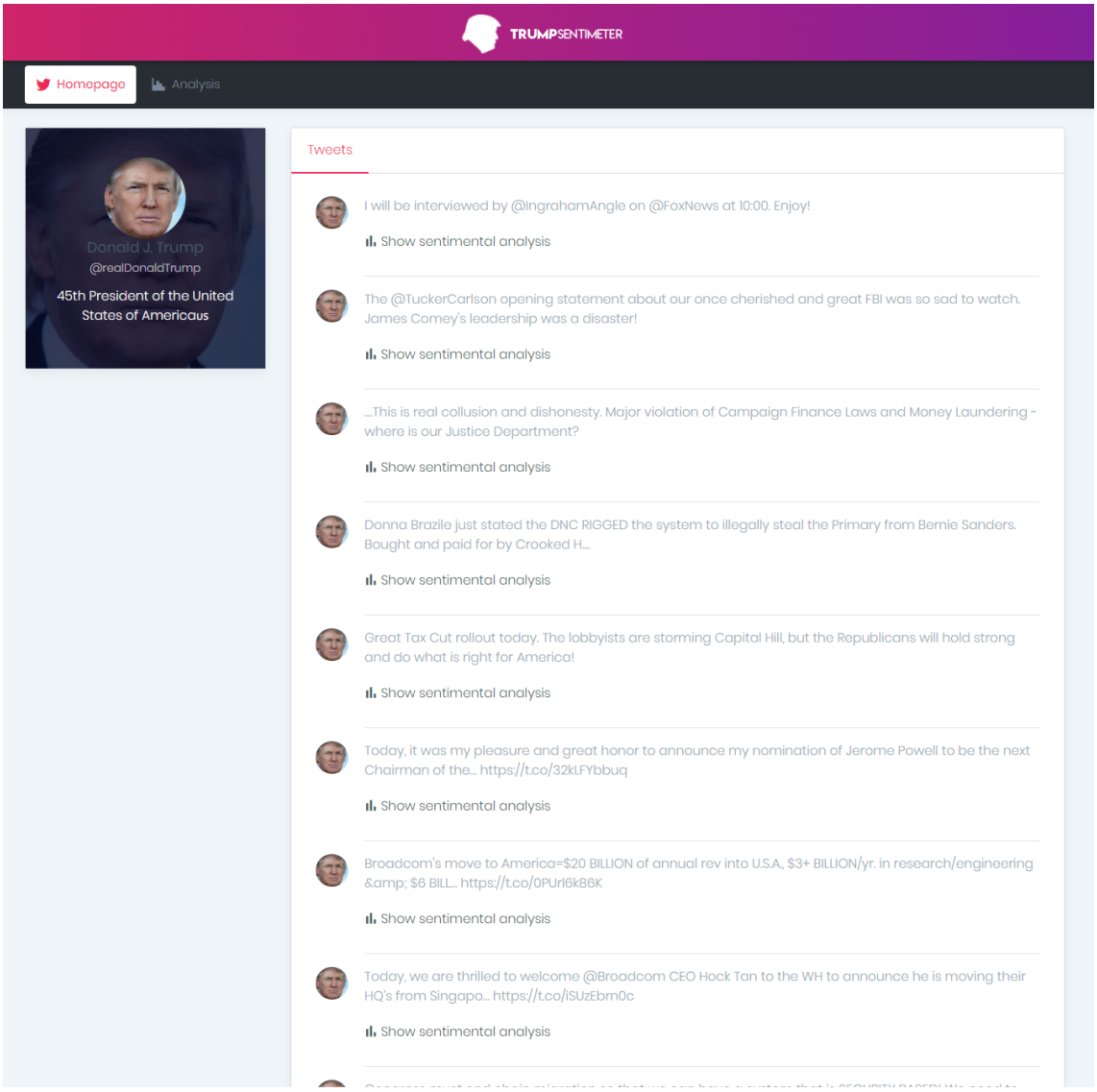

Figure 2: Homepage

5.3 Analysis

The analysis page shows the statistics of the latest 100 tweets (sentimental score, amount of retweets, amount of likes and a piechart showing the sentiment seperation). In figure 3 and 4 shows linecharts of these properties. What's very important to notice is that the sentiment doesn't decrease the amount of likes or retweets. Users are able to click on points in the graph to see more statistics on each tweet.

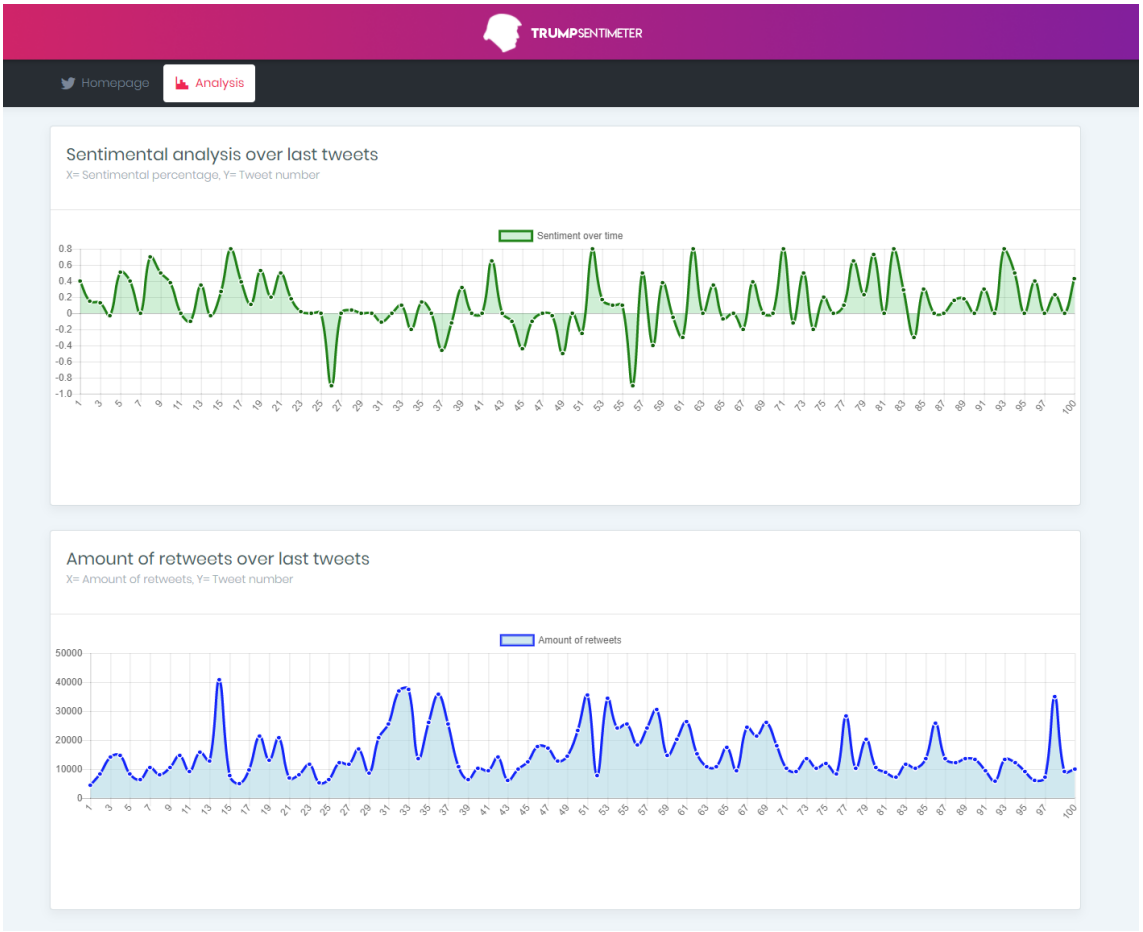

Figure 3: Sentimental analysis

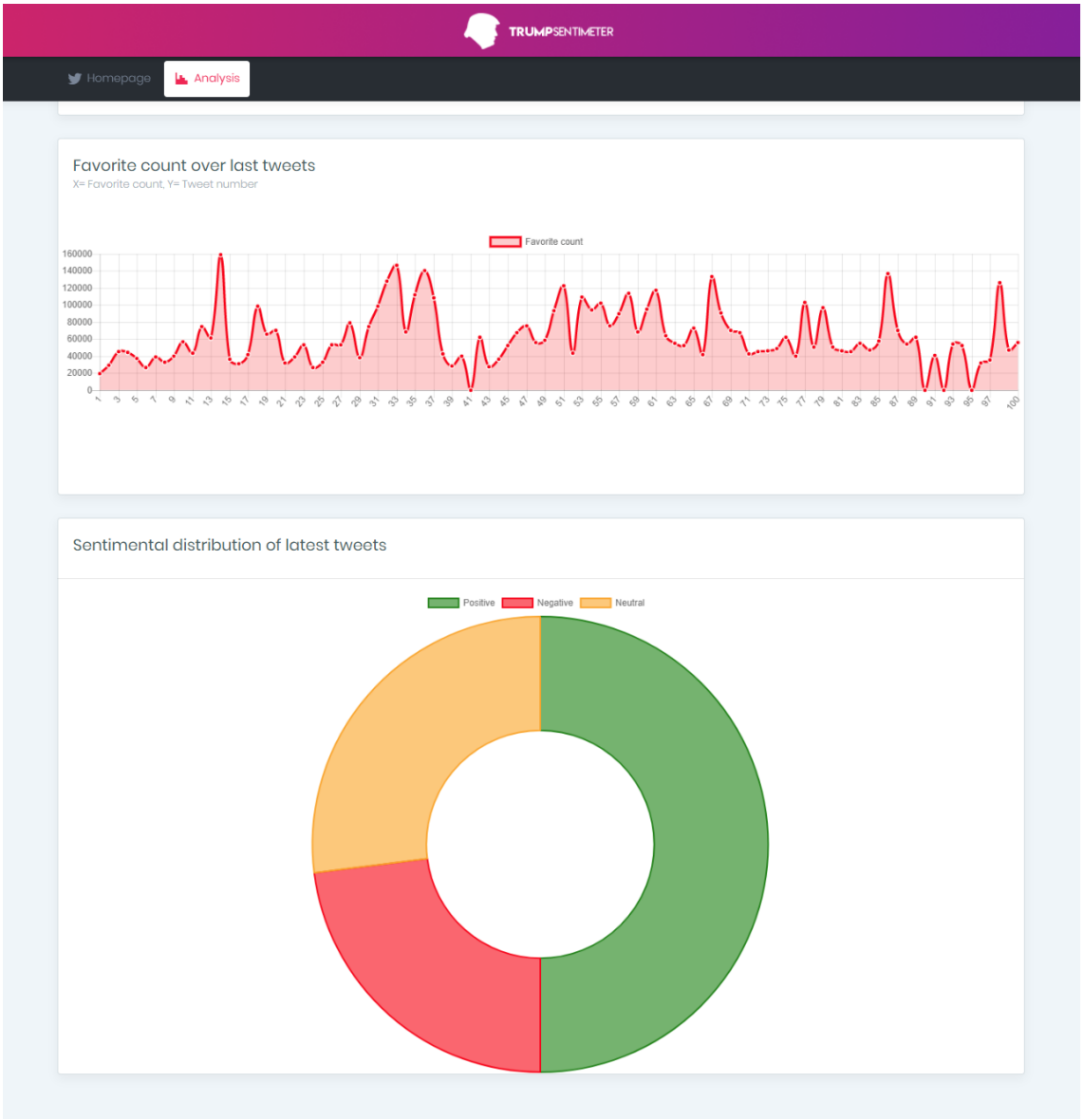

Figure 4: Sentimental analysis 2

5.4 Analysis by ID

The analysis by tweet id shows the most important information for sentimental analysis. The amount of retweets, likes and the sentimental score. See figure 5

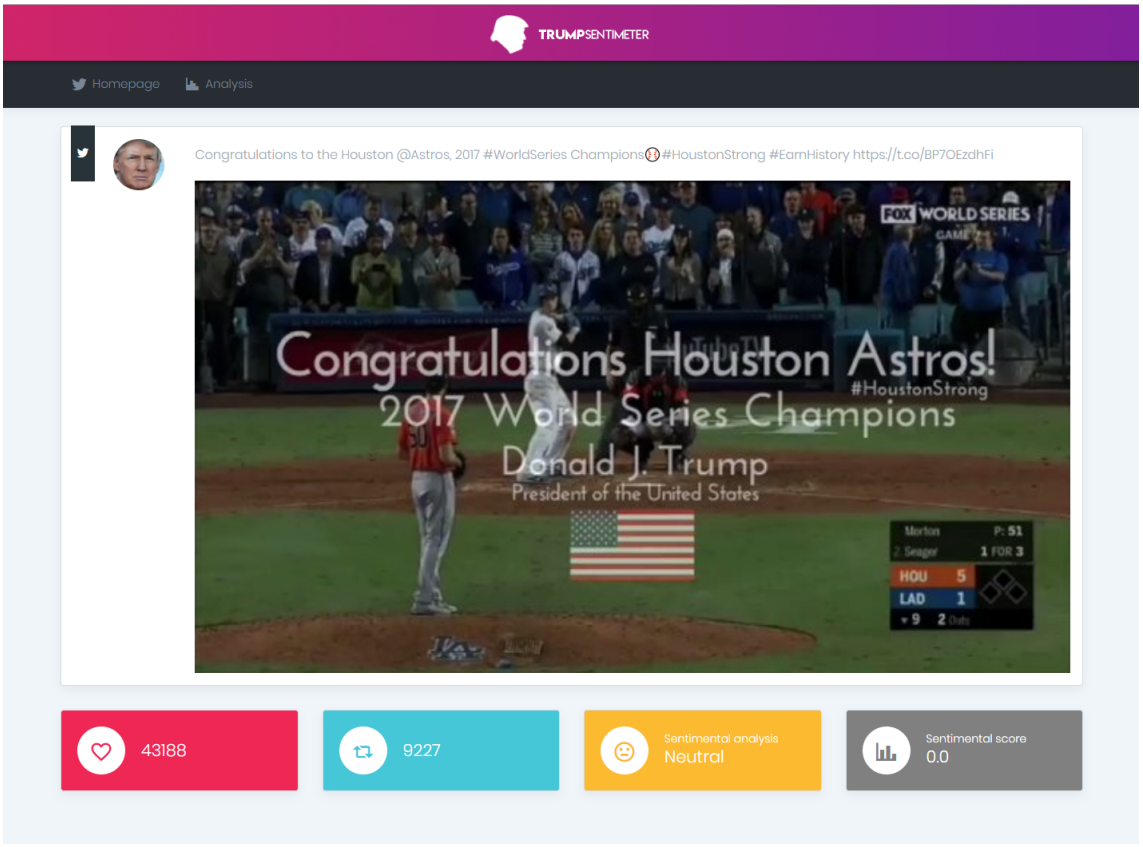

Figure 5: Analysis by Tweet ID

6 Evaluation

In general we are happy with the result of this project. In the beginning it was a bit unclear to us that the project was primarily focused on scalability. During the weeks we discovered that this was the main goal, which changed our view on the project. From that point we improved very fast and started focusing on the important parts of the project.

One of the key differences between us and other groups is the use of PHP (laravel framework). Most of the TA's were very skeptical about using PHP (laravel framework), with this project we hope that we have convinced you of using the laravel framework. Of course PHP has it's advantages and disadvantages in comparison to SCALA but the laravel framework fixes most of these issues.

The biggest problems we encountered during this project was with docker. The combination of laravel and docker was quite difficult, we would not recommend to combine these instances with the stack that was needed for this project.

Lastly we are happy with the result of the project, not only was it very fun to make (Making fun of Donald Trump). The site also has it's purpose of calculating a sentiment of tweets. It's also well written which gives us the opportunity to expand the project or change the person we want to analyse.