UNIVERSITY OF BERGEN

Department of Information Science and Media Studies

INFO319 PROGRAMMING PROJECT

Detecting hot topics in real-time using twitter streaming for situational awareness

Author: Candidates 2, 5, 9 and 12

November 1, 2019

Contents

1 Introduction

Big data is a very important asset in disaster management, yet big data is still relatively new. Therefore the technologies used alongside big data and emergency management is constantly evolving and finding new uses. Microblogging services, like Twitter, is also experiecing rapid growth and is being adopted by many[4], and its vast amount of UGC (User Generated Content) can be very valuable for disaster management applications. We have through this project proposed a system that processes a twitter stream and detects hot topics in real-time using big data technologies. The system consists of an application that filters, analyzes and visualizes the data to the user through a web based dashboard that displays the most frequently used words and hashtags using tag clouds. In addition to the tag clouds, the most recent tweets are also visualized with highlights on the relevant words contained in the tweet's content. We hope that this system can be used by professionals to monitor disaster prone areas for emergency prevention, or that it might inspire new and better work that is successfully put to use.

2 Data sources

We have used Twitter as our only data source for this project. We did this because they have easily available data through their open and public APIs (Application Programming Interface) as long as we had access tokens, which we got from applying on the Twitter developer web pages. Twitter is a large social media platform that is used widely all over the world, and therefore offers large quantities of data both to free and premium users. For this project we figured that the free version of the API was more than enough for testing purposes.

The Twitter API requires the use of filters whenever you request data from the API. These are either language, locations based on bounding boxes, following certain users or tracking specific keywords. Only one filtering method is required to begin streaming the data. Initially we were planning on using a location filter to receive data from different areas around the world where the chance of emergencies were higher than average, however, we later found that the amount of location tagged data consists of such a small amount of the total incoming tweets since the users have to allow twitter to share their current location. Therefore, considering we wanted to get data related to emergency management, we filtered the incoming data based on emergency related keywords to receive more relevant data and avoid losing potentially emergency relevant tweets. For testing purposes we used "earthquake", "flood" and "wildfire" as our tracking keywords.

3 Tools used

We mainly used python as our programming language for this project. In addition to python we used a few libraries to make the system more reliable and steady. We used Tweepy to connect to the twitter streaming api. Tweepy is a python library that includes a number of functions for communicating with the Twitter api. We only used this library for Twitter streaming.

For data processing we used another python library called pyspark. Pyspark is a python wrapper for the Apache Spark framework. Spark is a big data technology that allows for 100x faster workloads using state-of-the-art DAG scheduler, a query optimizer, and a physical execution engine[1]. We found another python library called Afinn[6], which is a simple library used for calculating the sentiment score of a text.

For the web api we used a python library called Flask together with a plugin for socketIO connections. The frontend is created using a frontend javascript framework called react and is styled using a stylesheet library called bootstrap. We have included a complete ordered list of all the tools we used below:

1. Python 3.x
   - Tweepy - A python library for accessing the twitter-api.
   - Flask - A micro-framework for web-application backend.
     - socketIO - A flask package for handling socketIO connections
   - Afinn - A python library used to calculate a sentiment score for a text.
   - Pyspark - A library to use spark in python
2. Apache Spark - An efficient big data technology for processing data
3. Frontend
   - ReactJS A javascript framework for creation interactive one page web-application
   - Bootstrap - A stylesheet library used to create flexible and mobile friendly web interfaces

4 System architecture

The system we have created consists of four different modules that all serve a different purpose for the complete system. One could also consider these to be separate programs, considering that they all have to be run separately from each other.

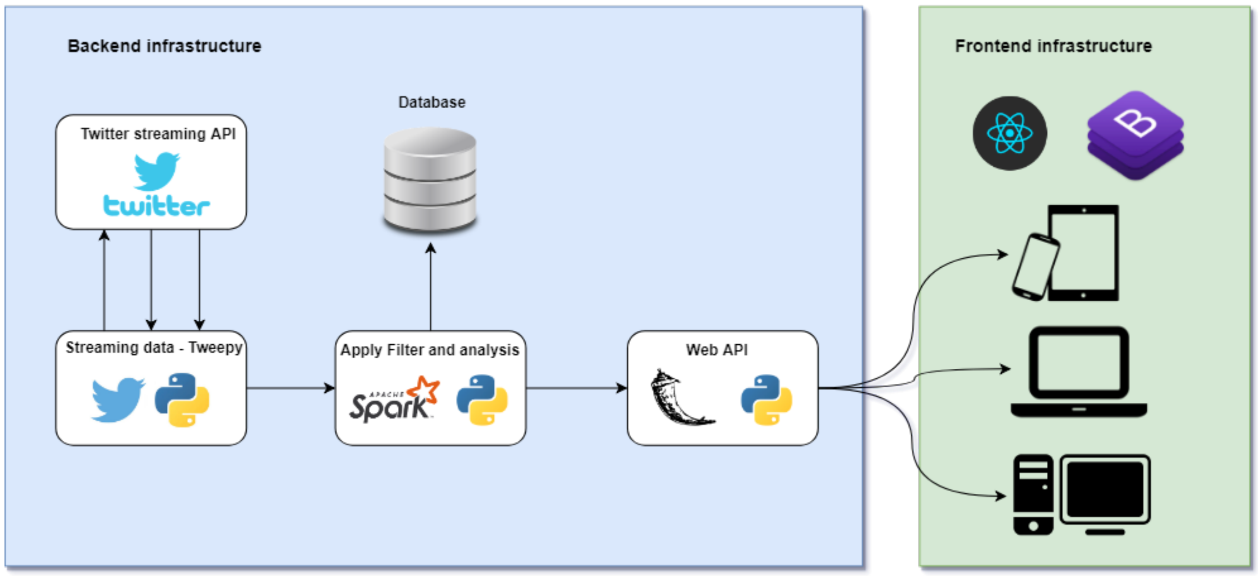


Figure 1: A model of the system architecture and data flow.

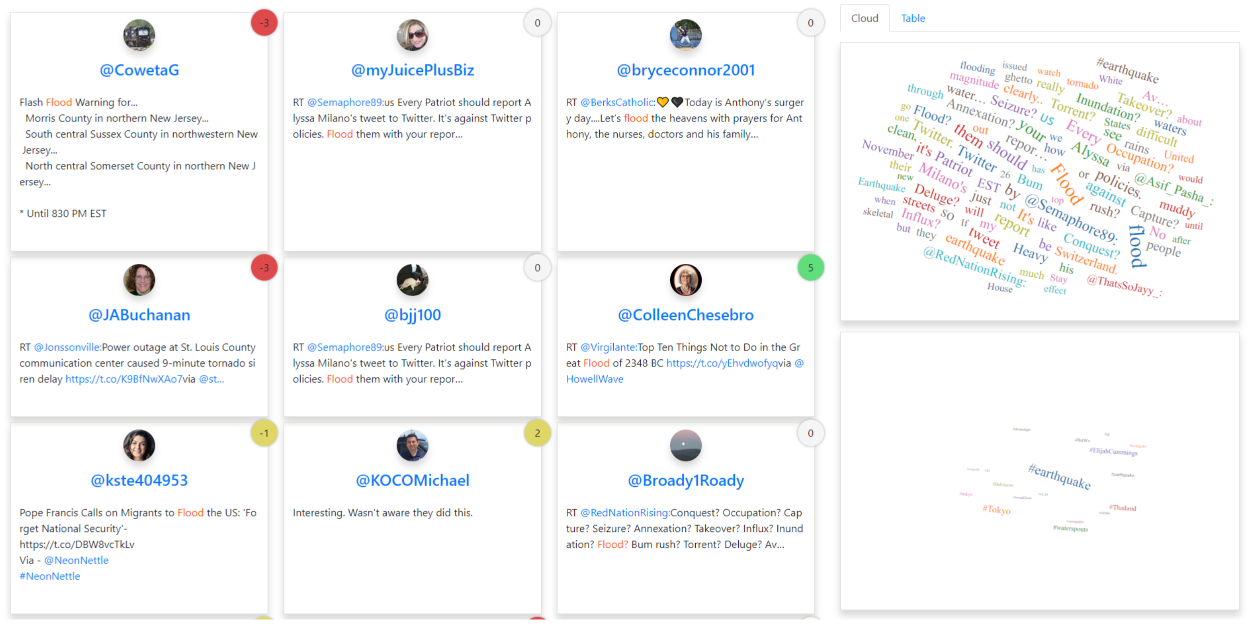
As can be seen in figure 1, we are first fetching data from the Twitter streaming api and then sending it along to Spark. Spark handles processing, analyzing and calculates the count of the current most used words and hashtags for a given time window. We will go into details of the analysis in section 4.1. Spark stores the data in a database to allow for later bulk processing. We used sql-lite for simplicity and testing purposes, however, in a more real scenario it could be more beneficial to use a database designed to be used for big data storage such as Cassandra or Hive. In addition to storing the data, it is sent to the web-api. The web-api only is only responsible for keeping control of connected clients and sending the calculated word and hashtag counts to the connected clients whenever it receives data from spark. The frontend consists of a dashboard created using reactJS and bootstrap, and displays the wordcounts using tag clouds and the related tweets displayed with their sentiment score.

4.1 Data analysis

The data analysis we did in Spark was mostly the calculation of the hot topics using the reduceByKeyAndWindow() function available in Spark. Before we could pass the data into the function we filtered out binding words and hyperlinks to avoid some of the unnecessary noise contained in the content. We did not do any keyword based filtering considering that the tweets we receive already have been filtered based on the keywords we have requested from the Twitter streaming api.

5 Dashboard

We wanted our dashboard to have a simple and structured design in order for it to be easy to use. For our dashboard we have used two main methods to visualize the content, tag cloud and news feed. The tag cloud is a visual display of hashtags, words, phrase or categories [2]. We made two widgets for the tag cloud, one displaying the most used word in the tweets and another for the most used hashtag. Our thought behind using the tag cloud is to highlight the most common words or hashtags present in a tweet so that it is easy for the user to see what is trending right now. The bigger the word or hashtag is displayed, the more it is mentioned or used. The other component of our dashboard is a news feed which consists of live tweets containing a specified set of keywords that we chose. The tweets also have a visual sentiment score attached to the top right corner.



-3
@CowetaG
Flash Flood Warning for...
Morris County in northern New Jersey...
South central Sussex County in northwestern New Jersey...
North central Somerset County in northern New Jersey...
* Until 830 PM EST

0
@myJuicePlusBiz
RT @Semaphore89:us Every Patriot should report Alyssa Milano's tweet to Twitter. It's against Twitter policies. Flood them with your repor...

0
@bryceconnor2001
RT @BerksCatholic:🧡🖤Today is Anthony's surgery day....Let's flood the heavens with prayers for Anthony, the nurses, doctors and his family...

-3
@JABuchanan
RT @Jonssonville:Power outage at St. Louis County communication center caused 9-minute tornado siren delay https://t.co/K9BfNwXAo7via @st...

0
@bjj100
RT @Semaphore89:us Every Patriot should report Alyssa Milano's tweet to Twitter. It's against Twitter policies. Flood them with your repor...

5
@ColleenChesebro
RT @Virgilante:Top Ten Things Not to Do in the Great Flood of 2348 BC https://t.co/yEhvdwofyqvia @HowellWave

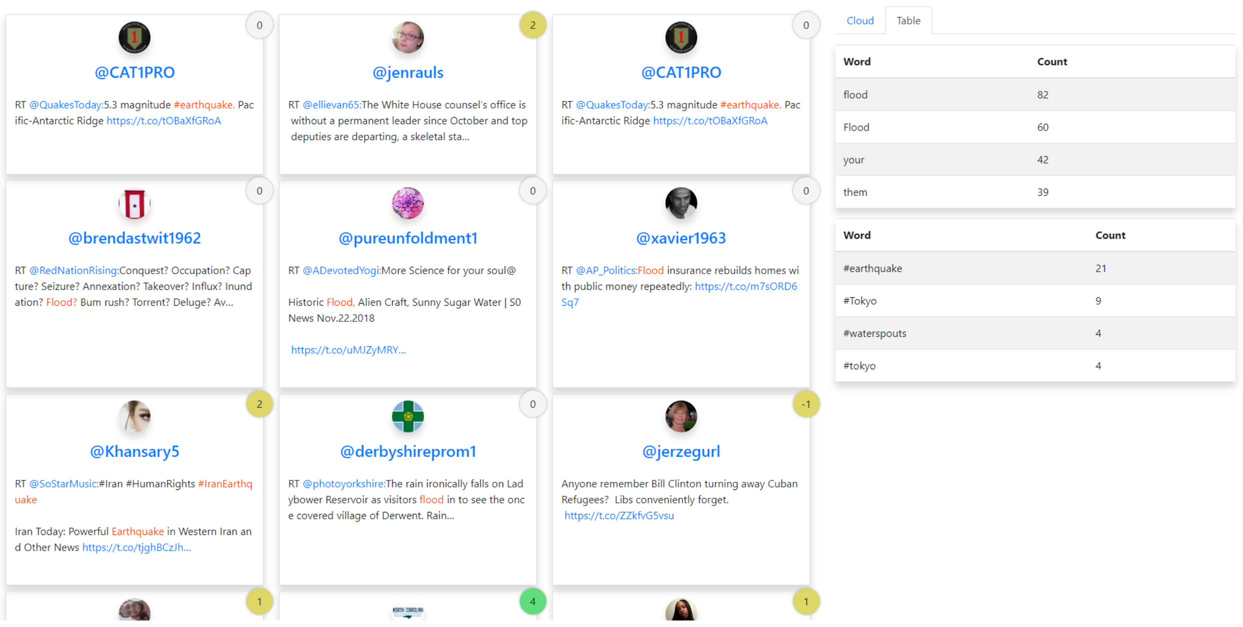
-1
@kste404953
Pope Francis Calls on Migrants to Flood the US: 'Forget National Security'-
https://t.co/DBW8vcTkLv
Via - @NeonNettle
#NeonNettle

2
@KOCOMichael
Interesting. Wasn't aware they did this.

0
@Broady1Roady
RT @RedNationRising:Conquest? Occupation? Capture? Seizure? Annexation? Takeover? Influx? Inundation? Flood? Bum rush? Torrent? Deluge? Av...

Figure 2: An example of the dashboard

An example of the dashboard that visualizes the data is showed in figure 2. The tag clouds are placed on the right side of the dashboard with the most used words being in the upper right corner and the most used hashtags on the bottom right corner. Above the tag cloud section there is a button where you can choose between a tag cloud or a list of the top four most used words or hashtags (figure 3). We implemented this in order for the humanitarian actor to be able to access the data even quicker if necessary. The newest related tweets are placed on the left side of the dashboard and is sorted from newest to latest. We have limited it to display the 50 newest tweets so that the browser can handle the amount of data streamed.



0
@CAT1PRO
RT @QuakesToday:5.3 magnitude #earthquake. Pacific-Antarctic Ridge https://t.co/tOBaXfGRoA

2
@jenrauls
RT @ellievan65:The White House counsel's office is without a permanent leader since October and top deputies are departing, a skeletal sta...

0
@CAT1PRO
RT @QuakesToday:5.3 magnitude #earthquake. Pacific-Antarctic Ridge https://t.co/tOBaXfGRoA

0
@brendastwit1962
RT @RedNationRising:Conquest? Occupation? Capture? Seizure? Annexation? Takeover? Influx? Inundation? Flood? Bum rush? Torrent? Deluge? Av...

0
@pureunfoldment1
RT @ADevotedYogi:More Science for your soul@

Historic Flood, Alien Craft, Sunny Sugar Water | S0 News Nov.22.2018

https://t.co/uMJZyMRY...

0
@xavier1963
RT @AP_Politics:Flood insurance rebuilds homes with public money repeatedly: https://t.co/m7sORD6Sq7

2
@Khansary5
RT @SoStarMusic:#Iran #HumanRights #IranEarthquake

Iran Today: Powerful Earthquake in Western Iran and Other News https://t.co/tjghBCzJh...

0
@derbyshireprom1
RT @photoyorkshire:The rain ironically falls on Ladybower Reservoir as visitors flood in to see the once covered village of Derwent. Rain...

-1
@jerzegurl
Anyone remember Bill Clinton turning away Cuban Refugees? Libs conveniently forget.
https://t.co/ZZkfvG5vsu

1

4

1

Word	Count
flood	82
Flood	60
your	42
them	39

Word	Count
#earthquake	21
#Tokyo	9
#waterspouts	4
#tokyo	4

Figure 3: An example showing the table data.

Tarasconi et al. proposed a system that groups tweets into different categories based on their content where one of them were an informativeness category[7]. We wanted to adapt a similar approach, showing only the most relevant tweets, but we did not manage to create an algorithm for this within the timeframe we were given. The system acts as a base for a larger system that could potentially be used for emergency monitoring and to gain a better situational awareness.

6 Existing solutions

As for existing solutions, we have found a couple of web applications that utilize a similar approach. An example of a system that we found to have the the most in common with is the web application ESA (Emergency Situation Awareness). According to their site they are a platform for monitoring real-time Twitter streams across the Oceania, to convert the deluge of data into situational awareness information for crisis coordinators [8]. And the information detected using their platform can be used to improve the understanding of the scope and impact of events such as earthquakes, cyclones, bushfires, and floods.[8]

Many web application is operational with real-time streaming, however, we have not found any platform which combines the real-time feed with visual sentiment analysis. The idea behind combining these two is to give the humanitarian actor an advantage when monitoring the Twitter stream, by making the sentiment analysis more visual by adding the score with a coloured circle. In this way they should be able to easily sort out which tweets that are more important than the others should a disaster occur. The red color represents negativity and the green color represents positivity in tweet contents. This way it is easier to distinguish good from bad, since we do not believe people talk positively about emergencies when it happens to them. Also in comparison to the design that ESA use, we argue that our design is tidier which again improves the usability for the humanitarian actors who are supposed to use the platform. Our system is an easily adopted free web application, making it more suitable for any volunteered organizations that want to help out with emergency prevention.

7 Future work

Initially we wanted to create more data visualization options for the dashboard, but we had to prioritize as we had to finish on time. We wanted to create a component that combines tag clouds, sentiment analysis and geographical maps. The component would display two layers, one layer visualizing tag clouds in the form of geographical maps which visualizes emerging topics and hashtags in a specific area. The second layer would display an average sentiment analysis score over said area, to visualize negativity and positivity in areas. The combined component would hopefully be very useful for the prevention stage of the DML (Disaster Management Model). We believe the technology we have used is sufficient to continue developing such components and appending them to the dashboard, and that the components for visualization could be very useful for constructing custom dashboards for emergency management applications. As for the current analysis and filtering methods we want to apply an improved noise removal algorithm and better relevance detection.

8 Conclusion

We made a simple and efficient dashboard for emergency management using big data visualization. The dashboard has no out-of-the-box customization abilities, but it is made using technologies that makes it very easy to expand and customize for specific needs. Since this is a web based dashboard, it will hopefully eliminate problems that come with platform specific applications and be easier to use by volunteered organizations.

References

[1] Apache. *Apache Spark*. Accessed 24-11-2018. 2018. URL: https://spark.apache.org/.

[2] Carlos Castillo. *Big Crisis Data: Social Media in Disasters and Time-Critical Situations*. Cambridge University Press, 2016. DOI: 10.1017/CBO9781316476840.

[3] Junghoon Chae et al. "Public behavior response analysis in disaster events utilizing visual analytics of microblog data". In: *Computers & Graphics* 38 (2014), pp. 51–60. ISSN: 0097-8493. DOI: https://doi.org/10.1016/j.cag.2013.10.008. URL: http://www.sciencedirect.com/science/article/pii/S0097849313001490.

[4] Amanda Hughes and Leysia Palen. "Twitter Adoption and Use in Mass Convergence and Emergency Events". In: *International Journal of Emergency Management* 6 (Feb. 2009), pp. 248–260. DOI: 10.1504/IJEM.2009.031564.

[5] Peter M. Landwehr et al. "Using tweets to support disaster planning, warning and response". In: *Safety Science* 90 (2016). Building Community Resilience to Global Hazards: A Sociotechnical Approach, pp. 33–47. ISSN: 0925-7535. DOI: https://doi.org/10.1016/j.ssci.2016.04.012. URL: http://www.sciencedirect.com/science/article/pii/S0925753516300546.

[6] F. Å. Nielsen. *AFINN*. Richard Petersens Plads, Building 321, DK-2800 Kgs. Lyngby, Mar. 2011. URL: http://www2.imm.dtu.dk/pubdb/p.php?6010.

[7] Francesco Tarasconi et al. "The Role of Unstructured Data in Real-Time Disaster-related Social Media Monitoring". In: *2017 Ieee International Conference on Big Data (Big Data)* (2017), pp. 3769–3778.

[8] Power et al. Yin et al. Cameron et al. *Dashboard for Emergency Situation Awareness*. Accessed 26-11-2018. 2014. URL: https://esa.csiro.au/aus/index.html.