Academic year 2017/2018

Universidad Complutense de Madrid

Facultad de Informática

Final Project

Aleksandra Zmudzinska

Blazej Wietczak

Analysis of Twitter activity by country during competitive events

Abstract

The subject of this work is an attempt to analyze data. Due to its popularity, availability, and the ability to comment on public events, it was decided to use Twitter data as data for analysis. Data obtained in this way are particularly interesting because of the large social and age cross-section that can be observed among Twitter users, so they give a fairly good statistical sample for a given location. The aim was to create a solution to present the number of tweets posted by users from different countries during media events. Geolocation is usually not included in the tweets, so it was decided to work on the location given by the user [1]. The main tools used in the work are the Twitter API, the non-relational database management system MongoDB and the Python high-level programming language. It was also necessary to use external databases to convert location to coordinates. Thanks to the use of solutions presented in the work, it is possible to present the activity of twitter users from different countries through different visualization methods. In order to analyze the data in the best way, it was decided to use the tweets chart in time, maps showing the total intensity of tweets for the 20 most popular locations for the entire event, and a movie based on maps in minute intervals. It was important for the results to be legible and visually appealing, understandable also for people not related to the project. For the analysis, it was decided to choose popular events on an international scale, the course of which will be possible to reproduce after a while during which it will be possible to observe certain culminating points. During the tests, it was checked how the project deals with two events, the Eurovision Song Contest and the UEFA Champions League match between Real Madrid and Paris Saint Germain, the analysis of the data obtained was also carried out and attempts were made to extract conclusions from them. Thanks to the proposed methods, it was possible to obtain very interesting results, among others, to observe how the culmination such as scoring a goal in a football match influences the activity of users on Twitter and for which nations the given event is the most important.

Keyword: Twitter, MongoDB, data cleaning, data-analysis, visualization, geo-location, bubble-map, chart, football match, Eurovision

Contents

1. Introduction

a. Twitter

According to Cambridge Dictionary, a social network is [2] : “a website or computer program that allows people to communicate and share information on the internet using a computer or mobile phone”.

Twitter [3] is a social network that provides a microblogging service. A registered user can send and read so-called tweets. A tweet [4] is a short text message (max. 280 characters) displayed on the author's profile and shown to users who are watching the profile [4]. Twitter allows tagging (using # in front of the word) and responding to other users (@ username = answer). Users write short messages in their Twitter profile via a website, SMS or via a mobile application.

Twitter [5] is fashionable among the elites and widly used by public figures, including by politicians, diplomats, publicists and journalists. Twitter [6] profiles also have newspapers, magazines, state institutions, companies, medical centers, celebrities, etc. Tweets of well-known personalities (e.g. the pope, the president of the USA) are often prepared by communication specialists.

New Internet communication channels, such as Twitter or Facebook, have helped to help demonstrators coordinate their activities and send information to the international community about these demonstrations [7]. Social networks help in communication, coordination and synchronization of manifestations

The term Twitter revolution may refer to:

- riots after the elections in Iran in 2009
- riots in Moldova in 2009
- revolution in Tunisia in 2010-2011
- revolution in Egypt in 2011

b. Tweets analysis

According to 2018 Global Digital suite of reports from We Are Social and Hootsuite now more than 4 billion people around the world using the internet [8].

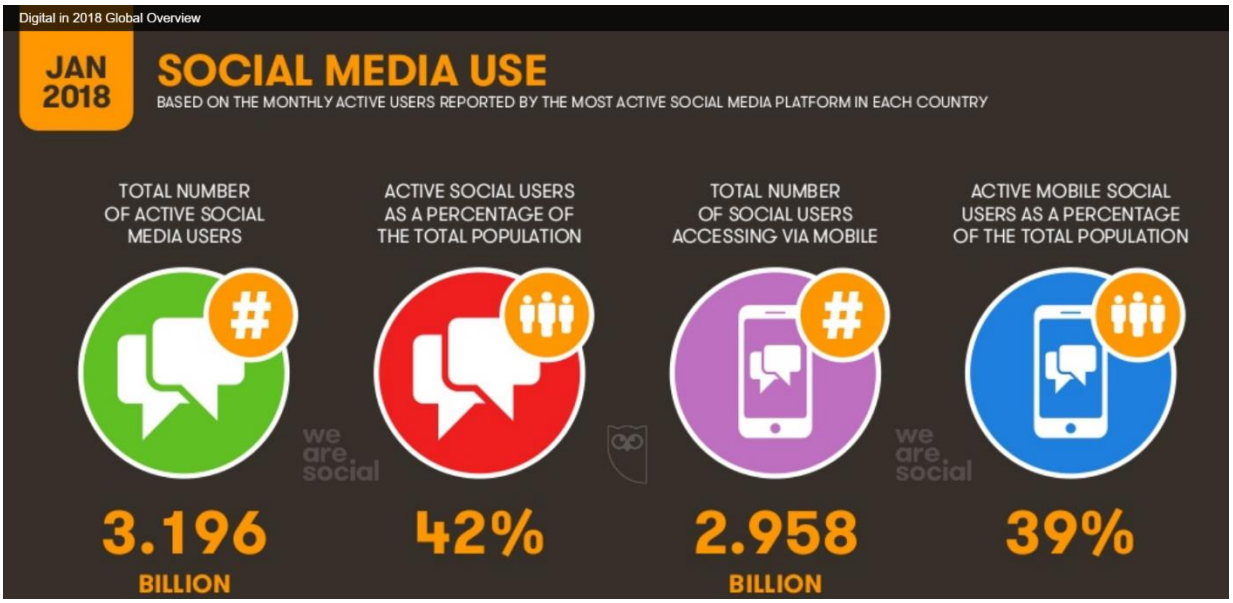


Figure 1 Statistics on the use of Social media

- The number of internet users in 2018 is 4.021 billion,
- The number of social media users in 2018 is 3.196 billion, which is 42% of all population,
- The number of mobile phone users in 2018 is 5.135 billion, which is 39% of all population,

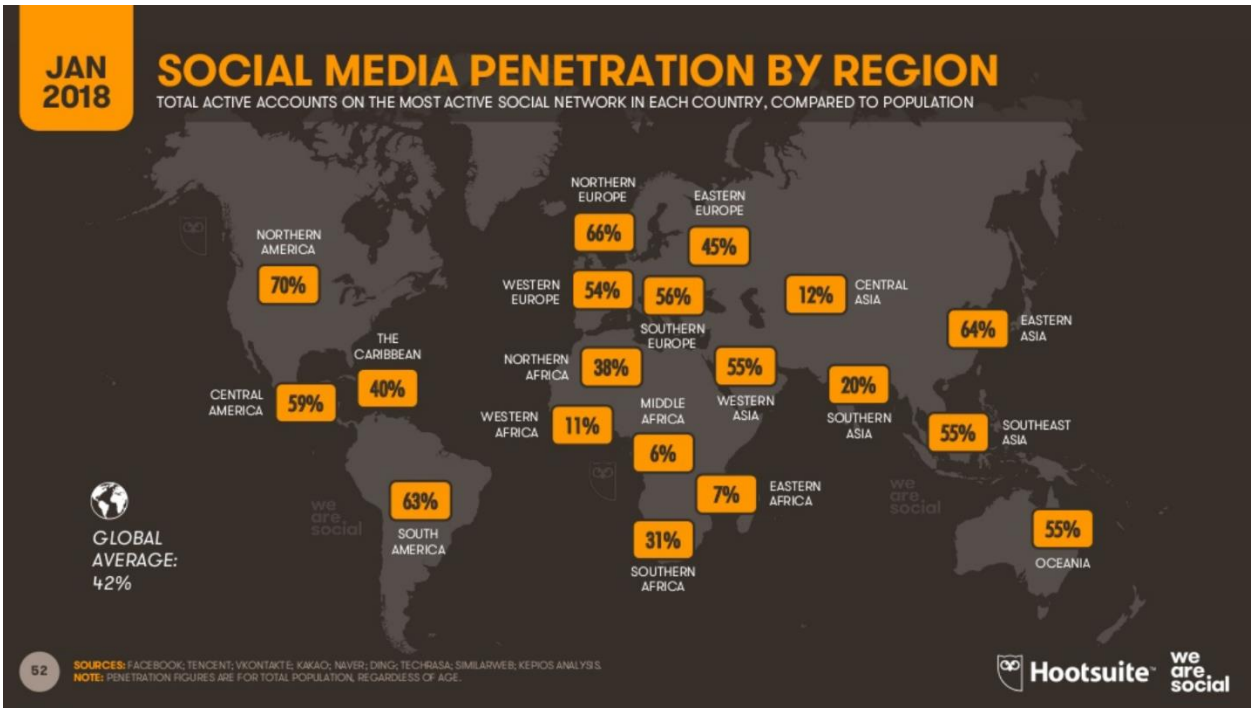


Figure 2 Map of the use of Social media around the world

Twitter is used by over 46% of the world's internet population, it allows commenting on events in the world in real time and watching tweets published by public figures, celebrities or politicians. This makes it a very powerful tool to observe political moods, social trends and what is popular in a given place in the world, in a given age group or for a given gender.

Since 2006, the possibility of using posts on Twitter as a source of sociological data has been tested. Software that is capable of processing natural language is tested for this purpose.

c. Examples of tweeter analysis (applications)

Data from Twitter is an interesting source of information about behaviors, interests and trends [9]. Many companies use information from it extensively, creating marketing or political campaigns, investigating the reactions of people to events or even examining the interest in specific cultural events. Using Twitter, it is possible to collect materials from any part of the world, almost immediately after they have been generated. The result is that in recent years many applications have been created that allow you to connect to the Twitter API and download data. Below are the most popular applications using data from the twitter site.

Table 1 The most popular applications using data from the twitter site

Name	Notes
BackTweets	Twitter monitoring tool which allows search for keywords in historical tweets. It also has an advanced search option which makes your searching more flexible.
Monitter	Monitor Twitter world for s set of keywords with a visual presentation of messages containing at least one of them.
Twazzup	It informs you every time your keywords are mentioned in a tweet. Categorizes results due to the popularity of messages, users sending them
TweetBuzzer	Shows you which brands is the most popular on Twitter. You can choose a 24 hour, 7-day, or 30-day period.
TwiBuzz	This tool tells you how often people are using Twitter to tweet your favorite keywords in real time.

The described applications allow various operations on data from Twitter [1]. Most of them are focused either on historical data or on tweets being processed at longer intervals [10]. However, no solution is known to allow analyzing the number of tweets and getting information about the place from where they were sent.

2. The aim of the project

The aim of the project was to show how the number of tweets on the twitter website changes during international events. The first stage was to collect all tweets about a specific event. Then clean and prepare the data so that only the information that interests us is preserved. The next step was to assign each tweet to geolocation in such a way that it would be possible to mark it on the map. The final part of the work was to prepare a presentation of the collected data using a video showing the change in the number of tweets sent by users from individual countries.

3. Programming environment

a. MongoDB

MongoDB [11] is an open, non-relational database management system written in C ++. It is characterized by high scalability, performance and the lack of a well-defined structure of supported databases. Instead, the data is stored as JSON-style documents, which allows applications to more naturally process them, while maintaining the ability to create hierarchies and indexing

b. Python

Figure 3 Python Logo

Python [12] is a high level general-purpose programming language, with an extensive package of standard libraries, whose guiding concept is the readability and clarity of the source code. Its syntax is characterized by transparency and brevity.

Python supports various programming paradigms: object-oriented, imperative and, to a lesser extent, functional. It has a fully dynamic type system and automatic memory

management, being like Perl, Ruby, Scheme or Tcl. Like other dynamic languages, it is often used as a scripting language. Python interpreters are available on many operating systems.

c. PyMongo

PyMongo [13] is a Python distribution containing tools for working with MongoDB and is the recommended way to work with MongoDB from Python.

Listing 1 Using PyMongo module

```
#!/usr/bin/env python
import pymongo
import time
import sys
import codecs

tim = []
sys.stdout = codecs.getwriter("iso-8859-1")(sys.stdout, 'xmlcharrefreplace')
connection = pymongo.MongoClient("mongodb://localhost")
db = connection.euro
loc = db.locations

def find():

    query = {}

    try:
        cursor = loc.find(query, no_cursor_timeout=True).batch_size(10)
        cursor.sort([('time_seconds', pymongo.ASCENDING)])

    except Exception as e:
        print
        "Unexpected error:", type(e), e

    for doc in cursor:
        time = doc['time_seconds']
        tim.append(time)

```

1. Implementation

a. Selection of the event

In order to get the most interesting results, we decided to choose an event that will interest audiences from different parts of the world. We decided that it would be good if it would be possible to observe moments that are more exciting for a certain group of recipients. The important thing was also to make it possible to compare the results with the course of the event. In the first place we decided to observe the football game,

where we have 2 teams from different countries that fight each other and observe the reaction of the users when climactic moments such as scoring, fouls appear. We can also follow the course of the match in a newspaper or on the website. We decided, therefore, that this is the perfect event for our project.

Another event that we have found that can give us interesting results is the Eurovision. It is a song contest, organized annually since 1956, in which 56 countries of Europe, North Africa and the Middle East take part [14]. It is a live competition, shown on TV in which teams from different countries compete. We expected to observe which countries or cities will be most interested in the competition, as well as the temporary activity of users from the country from which the performance is being presented.

As selected events, we used data from:

- Real Madrid - PSG match, which took place on April 6, 2018

- Eurovision, which took place on May 12, 2018

b. Preparing data

The first step during creating the project was reading the data from the twitter. Because the apps.twitter.com application interface allows you to generate an access token and a secret key for the application owner to read tweets, this option was used. To read the data, a program written in Python was used that enabled reading tweets according to the declared key and saving them in the json format. Then it was necessary to create a database in MongoDB, and to create a collection of previously obtained data files. At this stage, we already had all the necessary data in the collection, however to be able to develop analysis and visualization of the location of tweets, it was necessary to clean data.

```
"_id" : ObjectId("5ac3416b83e90358741f5a64"),
"created_at" : "Tue Mar 06 21:41:23 +0000 2018",
"id" : NumberLong("971138700565467136"),
"id_str" : "971138700565467136",
"text" : "BOŞ KONUŞMUYORUZ BURDA https://t.co/Ne6KeOd7rl",
"display_text_range" : [
        0,
        22
],
"source" : "<a href=\"http://twitter.com/download/iphone\" rel=\"nofollow\">Twitter for iPhone</a>",
"truncated" : false,
"in_reply_to_status_id" : null,
"in_reply_to_status_id_str" : null,
"in_reply_to_user_id" : null,
"in_reply_to_user_id_str" : null,
"in_reply_to_screen_name" : null,
"user" : {
        "id" : 1166293873,
        "id_str" : "1166293873",
        "name" : "OA",
        "screen_name" : "onurrx",
        "location" : null,
        "url" : null,
        "description" : "Galatasaray iyiyse, biz de iyiyiz.",
        "translator_type" : "regular",
        "protected" : false,
        "verified" : false,
        "followers_count" : 3307,
        "friends_count" : 1899,
        "listed_count" : 4,
        "favourites_count" : 8446,
        "statuses_count" : 2964,
        "created_at" : "Sun Feb 10 15:59:08 +0000 2013",
        "utc_offset" : 7200,
        "time_zone" : "Helsinki",
        "geo_enabled" : true,
        "lang" : "tr",
        "contributors_enabled" : false,
        "is_translator" : false,
        "profile_background_color" : "70E1F5",
        "profile_background_image_url" : "http://pbs.twimg.com/profile_background_images/665924195939815424/Y_MgxYUE.jpg",
        "profile_background_image_url_https" : "https://pbs.twimg.com/profile_background_images/665924195939815424/Y_MgxYUE.jpg",
        "profile_background_tile" : true,
        "profile_link_color" : "000000",
        "profile_sidebar_border_color" : "FFFFFF",
        "profile_sidebar_fill_color" : "DDEEF6",
        "profile_text_color" : "333333",
        "profile_use_background_image" : true,
        "profile_image_url" : "http://pbs.twimg.com/profile_images/885952397566251009/JGIoXkPC_normal.jpg",
        "profile_image_url_https" : "https://pbs.twimg.com/profile_images/885952397566251009/JGIoXkPC_normal.jpg",
        "profile_banner_url" : "https://pbs.twimg.com/profile_banners/1166293873/1520188758",
        "default_profile" : false,
        "default_profile_image" : false,
        "following" : null,
        "follow_request_sent" : null,
```

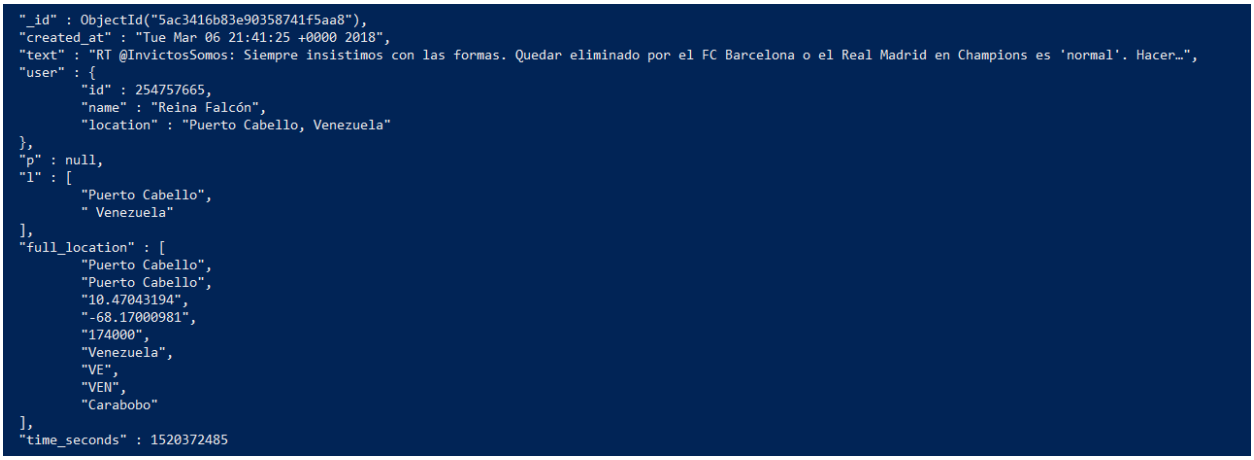

Figure 4 Screenshot of part of document of pure Twitter data

The first step in data cleaning was to check if the tweet has a location and delete those that do not have. Then it was necessary to remove unnecessary information and leave important such as: tweet creation time, tweet location, user id, tweet content.

The next step necessary to display tweets on the map was to obtain longitude and latitude for the given location. For this purpose, we checked the city for each tweet and searched for the city in the database containing the geographical locations of the cities. After finding the city in the database, we read the longitude and latitude and added it to our collection. The problem turned out to be that the database contained cities sorted according to the alphabetical order of the continents, for example the first city found was London in Africa and not in Europe, which introduced distortions in the data. To improve this, we sorted the data in the database containing the locations in terms of the

population descending and we repeated the process of adding longitude and latitude - thanks to this we obtained more reliable data.

Then we wanted to have the timestamp od each tweet in such a form to be able to sort after it and create time intervals. Tweeter ‘created_at’ is a String of UTC time when this Tweet was created. Example: "created_at" : "Tue Mar 06 21:41:23 +0000 2018". We decided to change it to time – tuple which contains 10 digits containing time with an accuracy of one second.



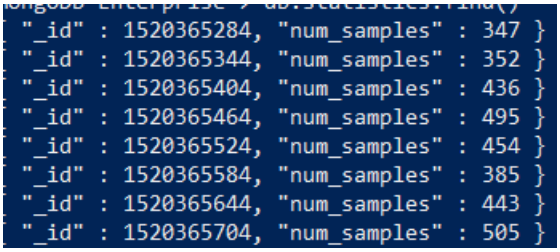
```
"_id" : ObjectId("5ac3416b83e90358741f5aa8"),
"created_at" : "Tue Mar 06 21:41:25 +0000 2018",
"text" : "RT @InvictosSomos: Siempre insistimos con las formas. Quedar eliminado por el FC Barcelona o el Real Madrid en Champions es 'normal'. Hacer…",
"user" : {
        "id" : 254757665,
        "name" : "Reina Falcón",
        "location" : "Puerto Cabello, Venezuela"
},
"p" : null,
"l" : [
        "Puerto Cabello",
        " Venezuela"
],
"full_location" : [
        "Puerto Cabello",
        "Puerto Cabello",
        "10.47043194",
        "-68.17000981",
        "174000",
        "Venezuela",
        "VE",
        "VEN",
        "Carabobo"
],
"time_seconds" : 1520372485
```


Figure 5 Screenshot of document from prepared data collection

c. Data selection

Once the data was properly cleaned and prepared, we could proceed to their selection. We aimed to create a chart, a map, and a movie.

The chart was supposed to show the number of tweets in time. For this purpose, it was necessary to aggregate the data over time, summing the number of tweets and sorting in time ascending.



```
{ "_id" : 1520365284, "num_samples" : 347 }
{ "_id" : 1520365344, "num_samples" : 352 }
{ "_id" : 1520365404, "num_samples" : 436 }
{ "_id" : 1520365464, "num_samples" : 495 }
{ "_id" : 1520365524, "num_samples" : 454 }
{ "_id" : 1520365584, "num_samples" : 385 }
{ "_id" : 1520365644, "num_samples" : 443 }
{ "_id" : 1520365704, "num_samples" : 505 }
```


Figure 6 Screenshot of data aggregated for chart

To create a map showing the number of tweets for a given location during the whole event, it was necessary to aggregate the data relative to the location, summing the number of tweets and sorting them descending. We decided to use the 20 most important locations, which was related to the limit function.

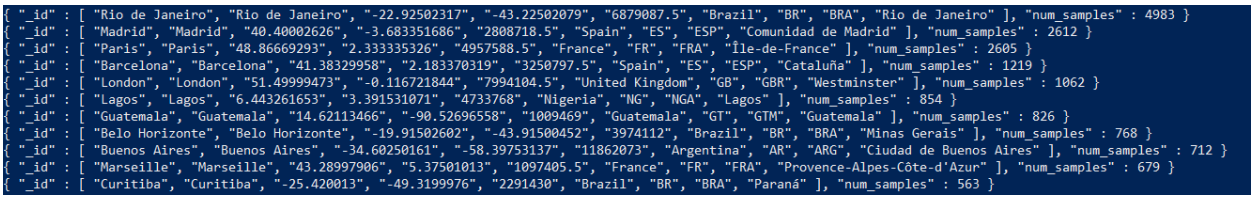


Figure 7 Screenshot of data aggregated for map

To create a movie, we decided to split the data in the time intervals every minute and for the given ranges to display the number of tweets for a given location. In this case, we also decided to limit the number of cities to 20. Selecting data for this purpose was a more difficult task.

```
query = {}

try:
    cursor = loc.find(query, no_cursor_timeout=True).batch_size(10)
    cursor.sort([('time_seconds', pymongo.ASCENDING)])

except Exception as e:
    print
    "Unexpected error:", type(e), e

for doc in cursor:
    time = doc['time_seconds']
    tim.append(time)


a = len(tim) -1
itv= (tim[a]-tim[0])/60
amin = tim[0] + 60
```

Figure 8 Part of the code which create time intervals and find the start and finish time

First, it was necessary to determine the time of the first and last tweet. Then specify the number of 60 seconds intervals by subtracting the start and end time and dividing by 60. Then, in the loop for time intervals, aggregations were made over the location, summing the number of tweets and sorting it descending. The data prepared in this way enabled visualization.

Figure 9 Screenshot of data aggregated for one interval for movie

d. Data visualization

To get the most information from our data, we decided to use several different methods of data visualization.

The first and the easiest way was the graph of the intensity of tweets from time. It allows you to easily observe whether an occurrence of an event - for example a goal, a break in a match, the performance of a team significantly affects the increase in the number of tweets. To show the data in the chart, we loaded them into the tables - time to one, tweets to the second and then we showed using the Python library - matplotlib in such a way that the time was on the x-axis, while on the y-axis was the number of tweets in the moment. To increase the readability of the chart, we decided to show data every 60 seconds.

```
time_array= []
tweet_array=[]
for x in range (0, len(timee),60):
    time_array.append(timee[x])
    tweet_array.append(tweets[x])

ax = plt.axes()
plt.plot( time_array,tweet_array, linestyle='-', marker='o')

plt.ylabel('total tweets number')
plt.xlabel('time in minuts')
plt.gcf().autofmt_xdate()
plt.savefig('chart.png')
plt.show()
```

Figure 10 The part of the code that accompanies adding elements to the array every 60 seconds and creating a graph

The next visualization method was a map showing the number of tweets for a given city using bubbles of various sizes during the whole event. Such a map made it possible to show which cities most actively tweeted about a given event. To display the data on the map, we used the Python - Folium library [15]. Using the aggregated data, we have loaded its latitude and longitude for each of the 20 cities as location, name of a city as a name and the number of tweets as a weight. First, we declared an empty map, giving parameters such as the center point and zoom. Then in the loop we gave the parameters of the next bubbles. By providing the number of tweets multiplied by the appropriate weight as the radius of the bubble, we get a larger bubble on the map the more tweets for the given location.

```
# Make an empty map
m = folium.Map(location=[50, 20], tiles="Mapbox Bright", zoom_start=4)

# I can add marker one by one on the map
for i in range(0, len(lat)):
    folium.Circle(
        location=[ float(lat[i]),float(lon[i])],
        popup=name[i],
        radius=(value[i] * 100),
        color='crimson',
        fill=True,
        fill_color='crimson'
    ).add_to(m)

# Save it as html
m.save('heatmap.html')
```

Figure 11 The code part which create an empty map, add bubbles and save the map as HTML file

At the end, the map with bubbles is saved as an html file.

The last method for visualizing the data we have carried out was a movie composed of maps displaying the number of tweets for a given location in minute intervals. This way of presenting the data allows to observe how people react from given regions of the world to a given situation during the entire event. This allows you to see if there are situations in which cities that were not generally active temporarily become active in response to some event. This part was done in a similar way to the previous one with the difference that we used data aggregated in time intervals and the whole map creation was done in a loop. First, the number of time intervals that determined the range of the loop was calculated.

```
db = connection.euro
loc = db.locations
query = {}

try:
    cursor = loc.find(query, no_cursor_timeout=True).batch_size(10)
    cursor.sort([('time_seconds', pymongo.ASCENDING)])

except Exception as e:
    print
    "Unexpected error:", type(e), e

for doc in cursor:
    time = doc['time_seconds']
    tim.append(time)

a = len(tim) - 1
itv = (tim[a] - tim[0]) / 60
print (itv)



for x in range (1,itv+2):
    db = connection.euro
    interval = db['interval'+str(x)]
```

Figure 12 The part of the code which obtain interval number and connect to collections with data aggregated in intervals

e. Automation

To assemble the movie from the created html files it was necessary to save them as a photo. Because we had approximately 250 html files, opening each of them separately and taking a screenshot would take a lot of time, we decided to automate this process. For this purpose, we used the Python library - Selenium and Chromedriver. Selenium WebDriver is one of the most popular tools for Web UI Automation. It can automate execution of the actions performed in a web browser window like navigating to a website, filling forms that include dealing with text boxes, radio buttons and drop downs, submitting the forms, browsing through web pages, handling pop-ups and so on. Chromedriver is a tool which allows automatic opening of files in the Google Chrome browser. The created program loads the html files in the loop, opens them in the Google Chrome browser, and then creates and saves the Screenshot of open on in the website map as a .png file using driver.save_screenshot() formula.

```
options = webdriver.ChromeOptions()
options.add_argument('--ignore-certificate-errors')
options.add_argument("--test-type")
options.binary_location = "/usr/bin/chromium"
driver = webdriver.Chrome("chromedriver.exe")

driver.get("file:///E:/Erasmus/projekt/eurovision/heat/" + str(i) + "heatmap.html")
driver.save_screenshot("screenshot" + str(i) + ".png")

driver.close()
```

Figure 13 The part of code which open the html file create a screenshot and save it into .png file

In this way, we get files ready to be assembled in a very short time.

f. Adding Tweets to movie

The resulting video was clear, it showed how the activity of users from a given place in the world changed during the event. However, we wanted it to be more interesting and provide more information. So, we decided to add a signature to each of the screenshots containing the time and one example of Tweet from the most active place in the given minute. For this purpose, we used the Python PIL library which enables operations in the pictures, among others, adding text in the picture. First, in the loop we read the name of the city with the most tweets and the time interval for the given interval. Then, for the given time frame, we searched for one tweet for a user from a given city and added this tweet as an inscription on the map's Screenshot from the given interval.

```
for doc in cursor:

    created = doc['created_at']
    time = datetime.strptime(created, '%a %b %d %H:%M:%S +0000 %Y').replace(tzinfo=pytz.UTC)
    text = time.strftime('%d %B %Y %H:%M')
    tweet = '#'+doc['text']

    print doc
    textOnPicture = text
    textOnPicture2 = tweet
    image = Image.open('E:/Erasmus/projekt/eurovision/Screenshot/screenshot' + str(z) + '.png')
    font_type = ImageFont.truetype('arial.ttf', 26)
    draw = ImageDraw.Draw(image)
    draw.text(xy=(1, 600), text=textOnPicture, fill=(0, 0, 0), font=font_type)
    draw.text(xy=(1, 660), text=textOnPicture2, fill=(0, 0, 0), font=font_type)
    image.save('m_image' + str(z) + '.png')
```

Figure 14 The part of the code which create the inscription to the map's Screenshot

By adding time to photos, we were able to compare users' activity with existing events, and by adding tweets, we could see what was the subject of user interests.

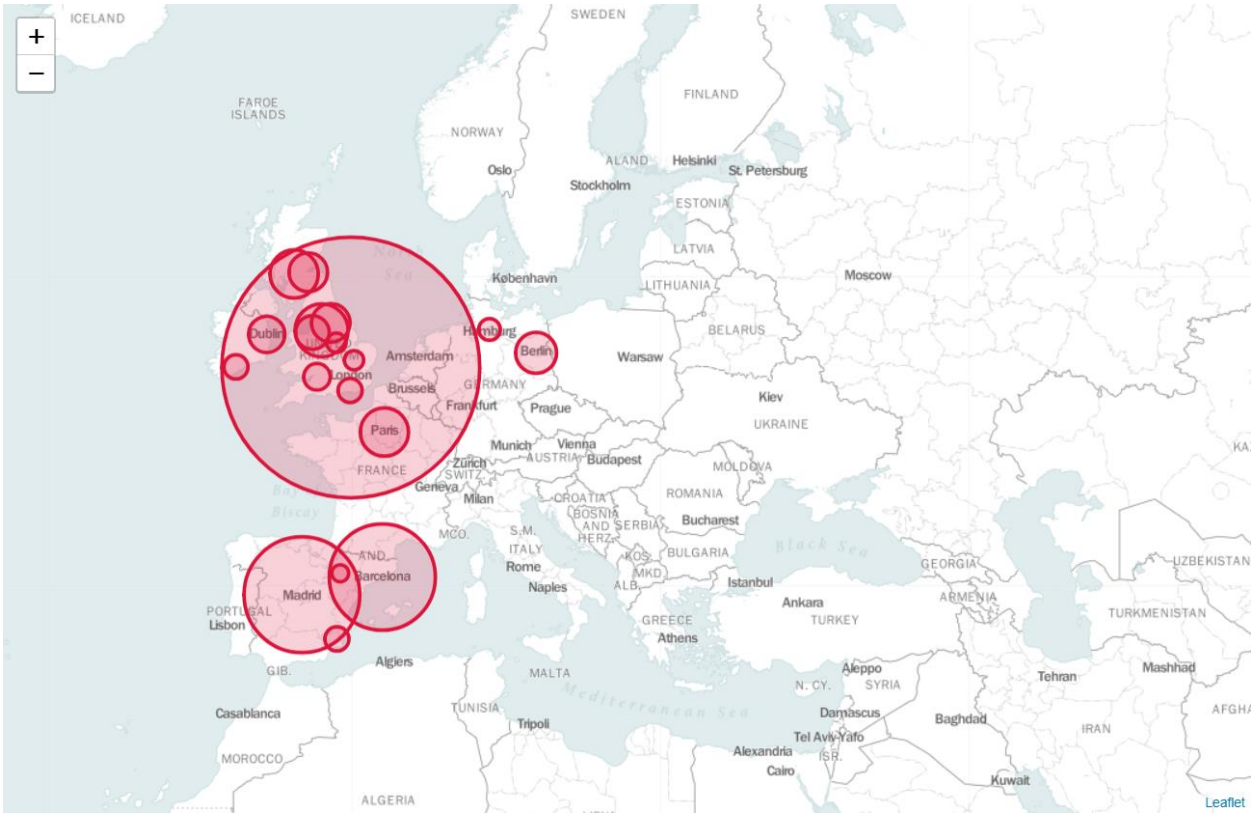


Figure 15 The map's screenshot without inscription

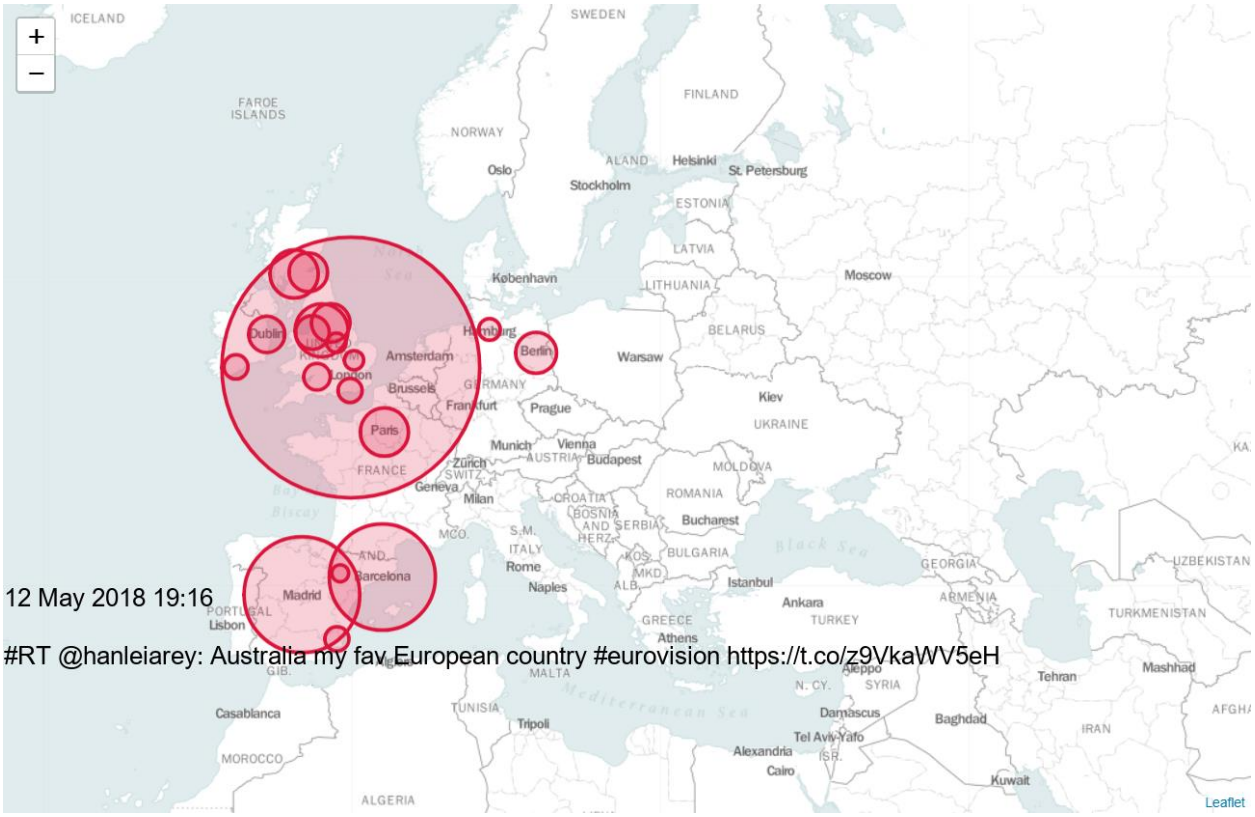


Figure 16 The map's screenshot with inscription - date, hour and tweet text

2. Results

We used two example datasets to check the project. The first of them was the Real Madrid - PSG football match, which took place on April 6, 2018, while the second one was the Eurovision contest, which took place on May 12, 2018. The aim of the project was to observe how the activity of twitter users in the world changes in response to the given event. To this end, we created a set of tools for visualizing results from each event.

a. Real Madrid - PSG football match

i. Chart

To illustrate how the general activity of twitter users changes during the football game, we created a chart that displays the number of tweets on the y-axis and the time on the x-axis. The timing of tweets starts a few minutes before the match and ends a few minutes after the match

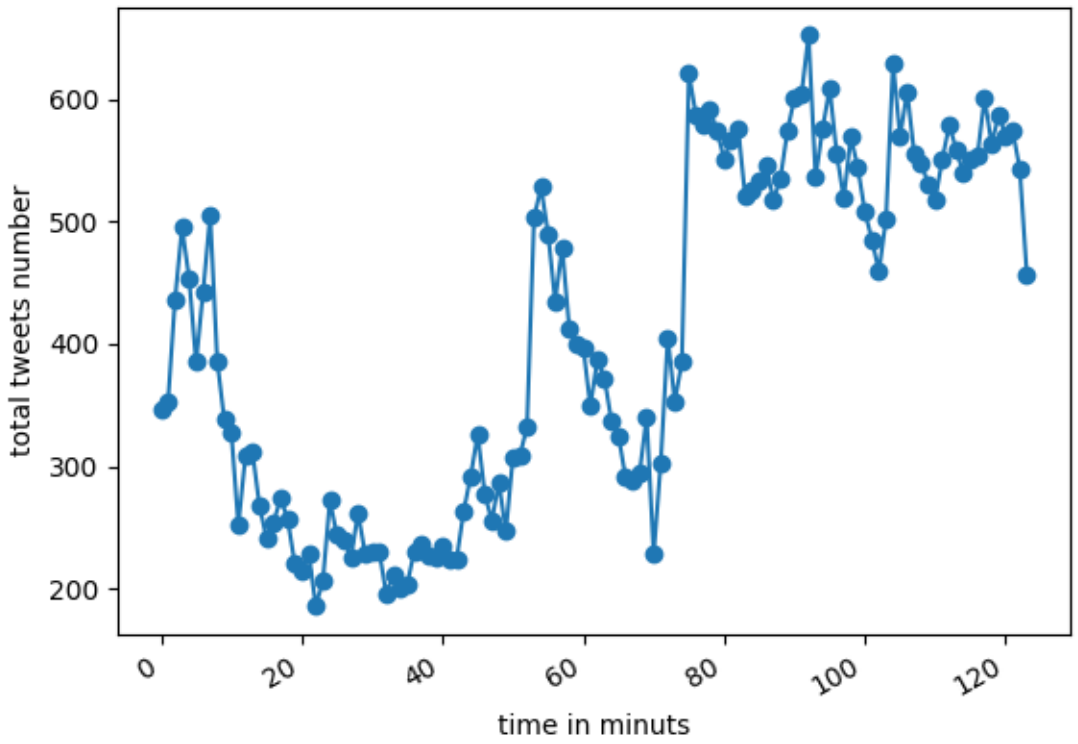


Figure 17 Chart of tweets number during match time of Real Madrid-PSG football match

The graph shows the high activity of twitter users a few minutes before the start of the match and during the beginning of the match. Later, you can see a great drop in the activity of twitter users. We can see small peaks at the 30th and 40th minute of the measurement, i.e. the 20th and 30th minute of the match, which corresponds to the yellow cards - first for Marco-Verratti (PSG), and then for Mateo Kovacić (Real - Madrid). Then, about 55-65 minutes of measurement, i.e. 45-55 minutes of the match, you can see increased activity. This is probably related to the break in the match - people who were previously busy watching the match, could add a tweet on twitter. Then you can see a drop-in activity and a very large increase of 75 minutes of measurement - this coincides with the scoring of the 51st minute by Cristiano Ronaldo (Real - Madrid). After this event, the activity of twitter users decreases slightly but still is on the high level. The next peak can be seen about 95 minutes, which can be interpreted as a reaction to the goal scored by Edinson Cavani (PSG). The last big increase in activity can be observed in the 105th minute, that is the 80th minute of the match, after scoring a goal by Casemiro and 10 minutes later after the match.

ii. Map

To see which cities were the most active during the whole match, we created a map with 20 places with the higher number of tweets in the entire match. The cities are surrounded by bubbles, the size of which is illustrated by the number of tweets - the bigger the bubble, twitter users from a given city were more active.

The map shows that the most active were users from Rio de Janeiro, Madrid and Paris, as well as from London and other cities from Spain and France. Interestingly, there is a lot of interest in Nigeria, Argentina, Peru, Venezuela, and several cities from Brazil outside of Rio de Janeiro

iii. Movie

The last way to present the data we used was a video created from the map screenshots showing the activity of users from different cities every minute. His goal was to observe how the activity of regions changes over time.

Figure 18 Screenshot of the movie

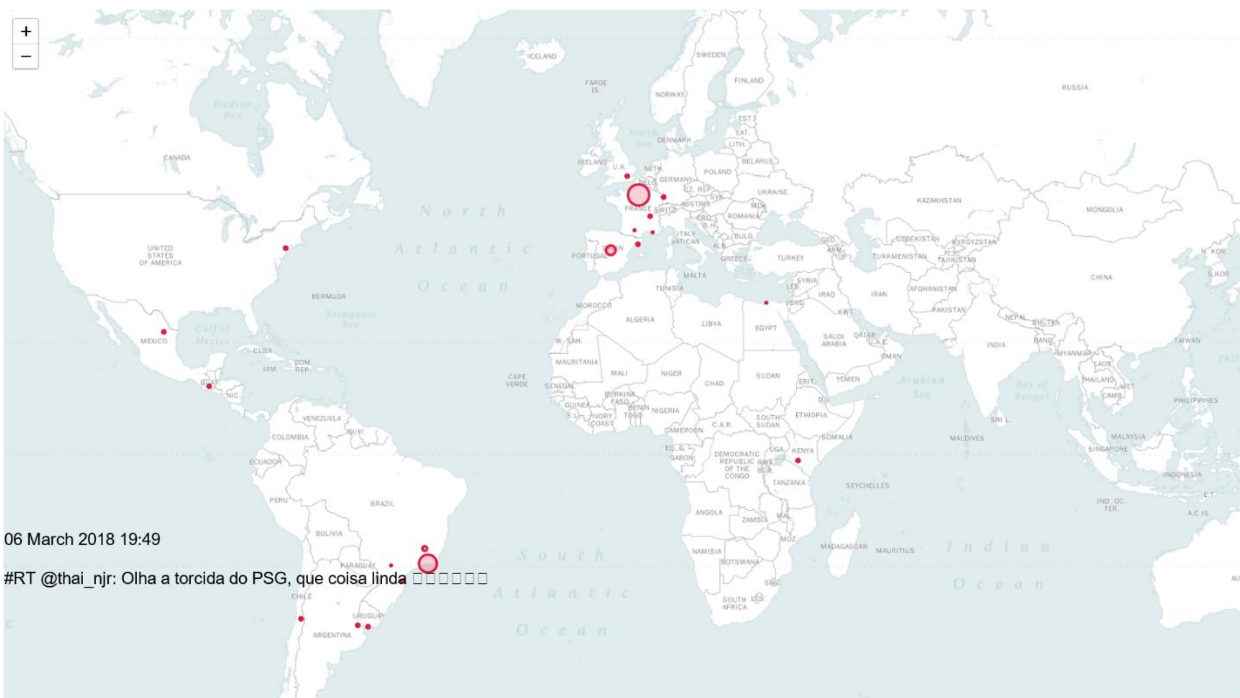


Figure 19 Screenshot of the movie

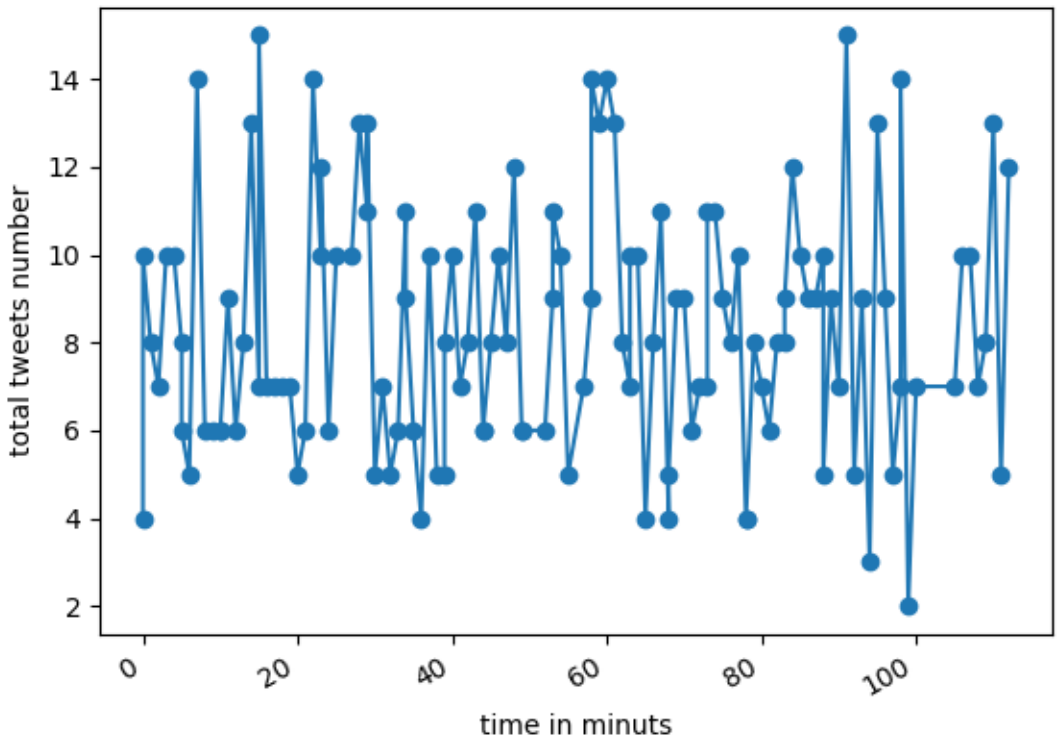
Figure 20 Screenshot of the movie

The attached examples clearly show that the activity of twitter users from a given place changes over time. We observed, among other things, that Rio de Janeiro and Madrid add a lot of tweets throughout the duration of the match while Paris and London were the most active during the break and at the end of the match

b. Eurovision

i. Chart

As in the case of a football game, the first step in analyzing Eurovision data was to create a chart of the number of tweets from time. We wanted to observe whether, as in the case of goals during the Real Madrid - PSG match, we will be able to observe increased activity during certain moments of the competition.



In the case of the Eurovision competition, we see much more even activity of tweeter users during the entire event. The number of tweets added is rapidly increasing every 10 minutes and then falls sharply. It is probably caused by adding tweets after the performance of a given vocalist.

ii. Map

Another method used to visualize the data was as in the case of the Real Madrid football game - PSG, showing the activity of given cities on the map during the entire event. On the map, we can see that twitter users from Great Britain, especially from London, were the most active. It is also possible to observe very high activity in Spain, with the most tweets coming from Madrid and Barcelona. You can also see a lot of activity in Berlin.

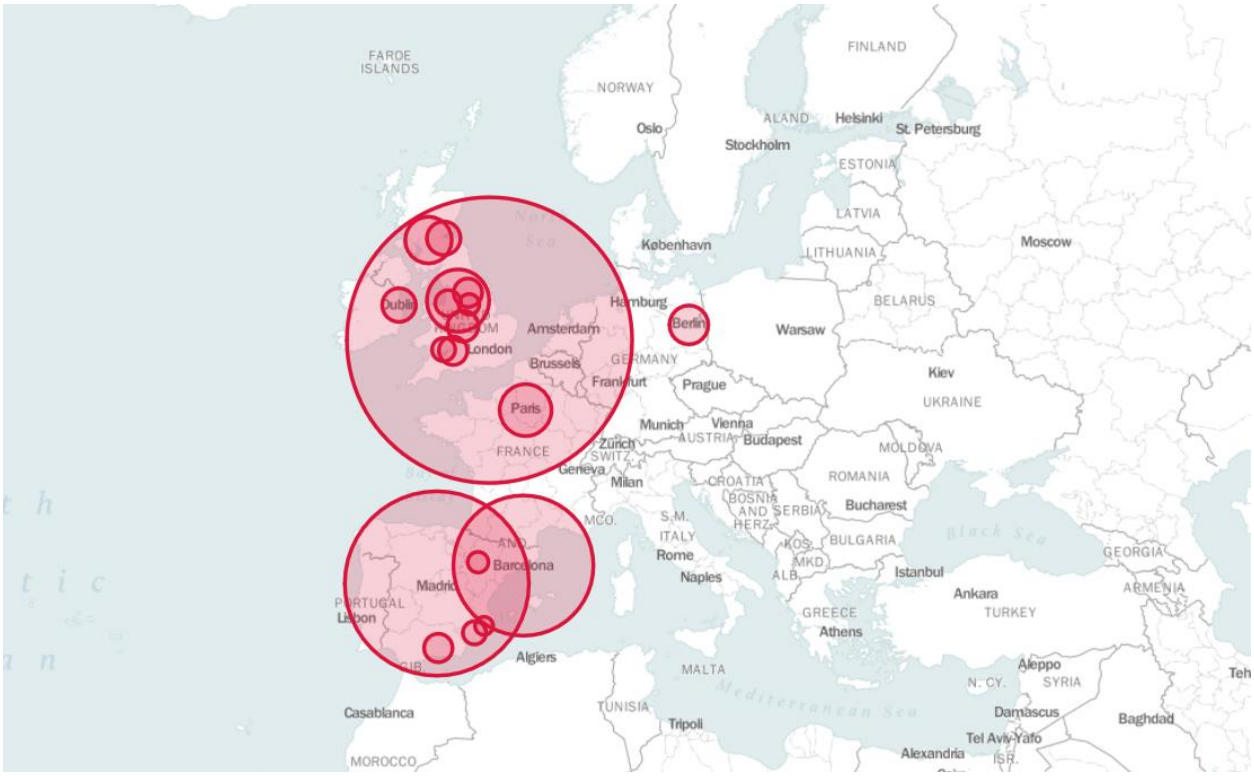


Figure 21 Heatmap of tweets during Eurovision

This is partly the case with viewers statistics which look as follows: the final was around 8.21 million viewers in Germany, about 7.2 million in Spain, about 6.9 million in Great Britain

iii. Movie

Just like in the case of a football game, we presented the obtained data in the form of a film, consisting of maps depicting the intensity of tweets for a given location data every minute. Just like in the case of a football game, we presented the obtained data in the form of a film, consisting of maps depicting the intensity of tweets for a given location data every minute. However, in the case of the Eurovision competition, there are much smaller changes in the activity of the cities in question compared to the data from the football game. Rather, one can notice a decrease in activity at certain times than a change in activity between cities.

Figure 22 Part of Eurovision movie

Figure 23 Part of the Eurovision movie - decreasing number of tweets observed

An interesting thing we noticed during the filming was a break in the data between 102 and 106 minutes. This was most likely caused by the disconnection of the connection to the network while reading data.

Figure 24 Part of Eurovision movie - break of data

3. Division of labor

The first part of the project was taken by Blazej. Acquiring data from the twitter site, data cleaning involving the removal of tweets from which it was impossible to obtain a location. Then add geolocation to the remaining tweets and change the time format in tweets.

The next part of the work was done by Aleksandra. It was the preparation of data so that you could create a chart, map and film from them. Then, visualize the data creating a chart and maps. Addition of subtitles (tweets) to photos and later automation of data aggregation.

The last stage of works belonged to Blazej. It consisted in creating png files from html files presenting maps and then automating this process and assembling a film from photos.

The presented division of labor is not one hundred percent correct. It is impossible because during the project implementation many of its functions and problems that appeared have worked together.

4. Conclusions

Twitter is a social network that provides a microblogging service. A registered user can send and read so-called tweets. A tweet is a short text message (max. 280 characters) displayed on the author's profile and shown to users who are watching the profile. The goal of the project was to analyze the data on media events from the twitter site with special consideration of how the tweets intensity in a given place in the world is shaped, as a response to the event and presenting the data graphically. Twitter data as data for analysis was decided to use, due to its popularity, availability, and the ability to comment on public events.

The following steps were taken during the creation of the project:

Media events were selected that could engage users from different countries around the world and increase user activity as a response to certain climaxes. It was also important to choose the media events which events whose course could be traced over time e. j. in the newspaper. It was decided to use two media events to be able to compare results, check the correctness of the program's operation for another data set and test the effectiveness of the program's automation. It was decided to use the football game (Real Madrid - PSG) due to the presence of clear climax such as scoring, foul, red card. The football game also had an advantage in the form that we could speculate that certain areas will be more active (the city from which teams come from and the countries from which the players come from) and on this basis, assess the correctness of the analysis. For example, if the results were obtained that during the Real Madrid (Madrid, Spain) match with PSG (Paris, France) the most active users were Poles and Hungarians, we could speculate that the program had errors when assigning locations from an external database. The second event that was decided to use was Eurovision. It was an interesting event due to its international scale and popularity among young people (the most-used age group from Twitter). This event did not have such characteristic points as the football match, so it could have brought some additional insights, or will there be a difference in the tweeting intensity between the two events. During such an event it seems likely that during the performance of the team from a given country, the increase activity of users from this country will be visible, so they would like to see if such relationships will be visible during the analysis.

Data were cleaned and prepared so that they were best suited for further analysis. It was necessary to delete many unnecessary data. You also had to choose only documents that contained the location of tweets. Then it was necessary to check whether the location really exists, or is not an artificial location, eg "Neverland". The next step was to add coordinates to each location so that it can be displayed on the map. For this purpose, it was necessary to use an external database containing: city, state, continent, longitude, latitude and population. Because the database was sorted

alphabetically to the continent, there were errors, and for example, the first London being read was London in Africa instead of in Europe. For this reason, it was decided to sort the database against the population. This approach resulted in very good results.

Data has been aggregated for every specific visualization method. The chart was supposed to show the number of tweets in time. For this purpose, it was necessary to aggregate the data over time, summing the number of tweets and sorting in time ascending. To create a map showing the number of tweets for a given location during the whole event, it was necessary to aggregate the data relative to the location, summing the number of tweets and sorting them descending. It was decided to use the 20 most important locations. To create a movie, it was decided to split the data in the time intervals every minute and for the given ranges to display the number of tweets for a given location. In this case, it was also decided to limit the number of cities to 20. This required the creation of a very large number of new collections, which would be very tedious if it had to be done manually and was the first step in automating the program.

Next, the data was presented by means of the tweet intensity graph over time, during the entire media event, which was to enable observation of whether at certain specific moments of the media event, users' activity may increase or decrease. Data was also present as a map showing the number of tweets for different locations during the entire media event, which aimed to show which areas were the most active during the entire event. Last kind of presentation data was a video showing the duration of the media event in minute intervals as maps showing the amount tweets for a given place in the world in a given time, which was supposed to show whether certain specific areas are activated during certain moments of the media event. This part also required automation, introduced automatic generation of maps in html format, automatic opening of the browser and taking screenshots, and then saving to PNG files. It was important for the results to be legible and visually appealing, understandable also for people not related to the project.

Thanks to the use of various methods, it was possible to observe both the activity of users during the media event and observe whether there were visible changes in the number of tweets during the climax of the media event, see which cities were most active during the entire media event and observe whether less active areas become active in response to any incident.

Based on the graph showing the football game, a clear increase in users' activity on scored goals was observed. It is very positive and confirms that the event was well-chosen and that this type of data visualization fulfills its assumption and is a valuable tool in analyzing data from Twitter. In the case of Eurovision, the number of tweets increases and decreases.

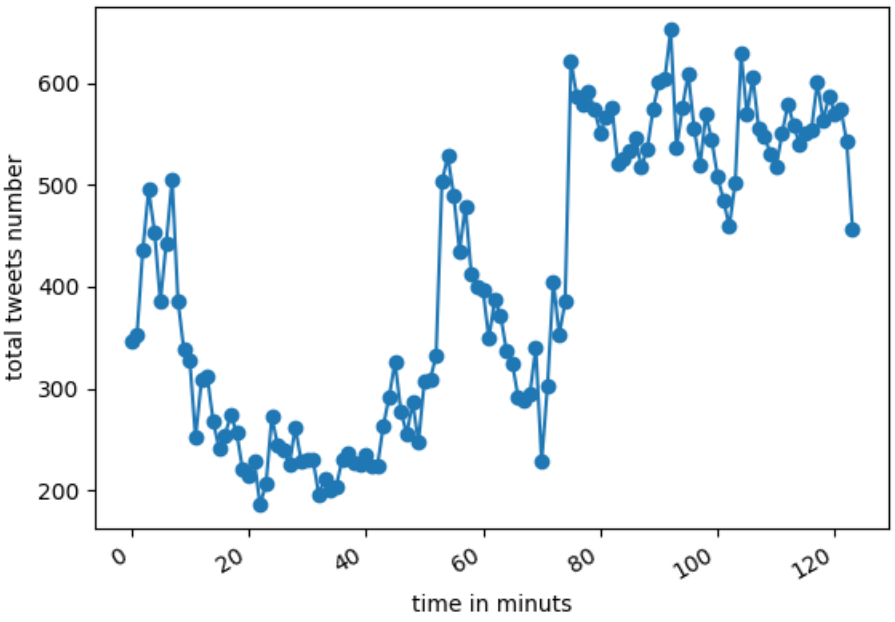


Figure 25 Football game - chart

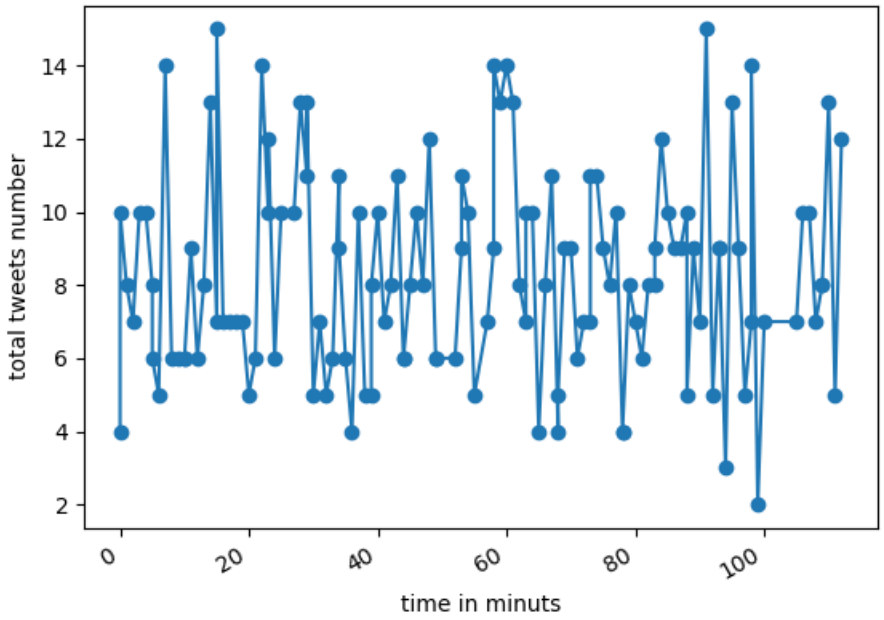


Figure 26 The Eurovision - chart

From the map depicting the location of tweets in a football game one can read that the most active users were people from Rio de Janeiro, Madrid and Paris, which is confirmed with speculations. During the visualization of Eurovision data using the map, we observed the largest number of tweets from Great Britain, Madrid and Barcelona. After checking the statistics of viewership, the convergence of results was noted.

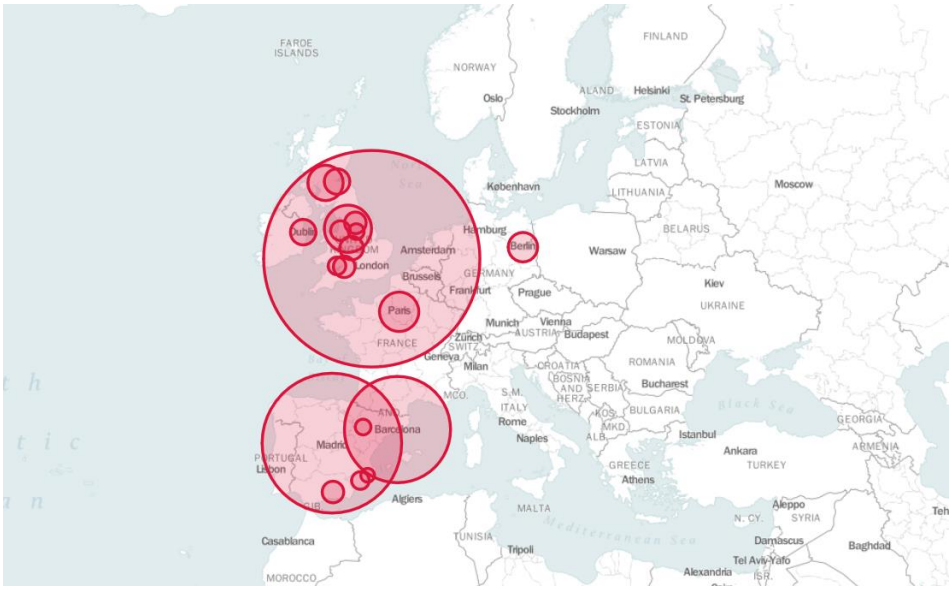


Figure 27 The Eurovision - map

Figure 28 Football game - map

The movie from the football game it is clearly visible that the activity of twitter users from a given place changes over time. We observed, among other things, that Rio de Janeiro and Madrid add a lot of tweets throughout the duration of the match while Paris and London were the most active during the break and at the end of the match. However, in

the case of the Eurovision competition, there are much smaller changes in the activity of the cities in question compared to the data from the football game. Rather, one can notice a decrease in activity at certain times than a change in activity between cities.

After analyzing the collected data, it became possible to draw the conclusion that the football game allowed for a more accurate observation of reactions from the viewers. We managed to observe that the increased number of tweets overlapped with important events during the match. It was also observed that during the movie from a football game, changes in the number of tweets for a given city are much more accurate than in the case of Eurovision. However, both events allowed us to observe which moments were most significant and which cities were most involved in commenting on the media event.

Even though we managed to present data from a twitter in a very interesting way, enabling data analysis, the project has the potential to introduce some corrections and to expand. An interesting possibility was to display the most common words for a given media event or even for certain time intervals. It would also be possible to study how the number of tweets in a given area is shaped depending on gender. The project could be implemented in the form of a windowed application, in which the user would specify the time range in which he wants to load files, e.g. on a website or a mobile application and receive the result in the form of appropriate charts and statistics generated fully automatically. However, if we would like to expand and transform the project into a commercial project, we would have to take into account the development rules included in [16]: https://developer.twitter.com/en/developer-terms/geo-guidelines.html.

The aim of the work was realized, we managed to create a project that allows the analysis of data obtained from the Twitter social network during particular media events.

5. Program files

Program files have been placed in google drive access via the following link:

https://drive.google.com/drive/folders/1dg_puyjBwUR8tPHC6Q7rNYUwvPrWbl6p?usp=sharing

6. Bibliography

[1] D. Liben-Nowell, J. Novak, R. Kumar, P. Raghavan, and A. Tomkins. Geographic routing in social networks. Proceedings of the National Academy of Sciences, 102(33):11623--1162, 2005.

[2] https://dictionary.cambridge.org/

[3] https://www.lifewire.com/what-is-a-tweet-3486211

[4] https://twitter.com/

[5] Akshay Java, Xiaodan Song, Tim Finin, Belle Tseng. Why we twitter: understanding microblogging usage and communities. San Jose, California — August 12 - 12, 2007

[6] https://www.telegraph.co.uk/technology/twitter/7297541/Twitter-users-send-50-million-tweets-per-day.html

[7] Alexandra Segerberg & W. Lance Bennett, Social Media and the Organization of Collective Action: using Twitter to explore the ecologies of two climate change protests.

[8] https://wearesocial.com/blog/2018/01/global-digital-report-2018

[9] Richard D.Waters, Jia Y.Jamal.Tweet, tweet, tweet: A content analysis of nonprofit organizations' Twitter updates. Public Relations Review Volume 37, Issue 3, September 2011, Pages 321-324

[10] Vanessa Murdock, Sheila KINSELLA. Locating a user based on aggregated tweet content associated with a location. US8478701B2 US Grant

[11] https://www.python.org/

[12] https://www.mongodb.com/

[13] https://api.mongodb.com/python/current/

[14] https://eurovision.tv/

[15] http://folium.readthedocs.io/en/latest/quickstart.html

[16] https://developer.twitter.com/en/developer-terms/geo-guidelines.html.