Politecnico di Torino

Corso di Laurea in Ingegneria Informatica

Tesi di Laurea Magistrale

Location learning and prediction in Social Networks

Supervisors

Prof.ssa Tania Cerquitelli
Prof. Carlos Garcia Rubio

Condidate

Pasquale Digiorgio

Torino, 2019

The world's most valuable resource is no longer oil, but data

The Economist, May 6th 2017

ABSTRACT

Nowadays, the analysis of data from Online Social Networks (OSNs) is one of the main areas of interest for companies involved in data analysis. A particular type of OSNs are Location-Based Social Networks (LBSN)s, which in addition to providing the normal functions of social networks add location-based services. This research aims to analyze the data of one of the most widely used LSBNs at the moment, namely Twitter, and calculate the entropy variation of the latter to identify any anomalies in different geographical areas. A careful analysis has been carried out on what is currently the panorama of data analysis made on the most used social networks at the moment. The result is the enormous difficulty in obtaining data from LSBNs due to the privacy restrictions imposed in recent years. In this project a code has been developed that allows to obtain in real time the geolocalized data coming from Twitter. Subsequently these data were analyzed and filtered to be subjected to different statistics, in this context was developed an interactive map in Python that allows to see the distribution of tweets in the different areas of the chosen geographical areas. The proposal is tested on a set of data collected by Twitter during a month in Madrid and compared with another set of data from Rome. Finally, an algorithm was applied to the data obtained for the calculation of entropy provided by the Department of Telematic Engineering of the University Carlos III, which allowed us to analyze the trend of the crowd during the period examined. Finally, this work aims to provide both the methods to obtain geolocalized data in Streaming from Twitter and those to analyze them carefully. Our goal is to find anomalies. The most obvious in a short trace in which nothing unusual happens is that the behavior on the weekend is different from the labour day, but the objective is to detect any other unusual behavior.

ACKNOWLEDGMENTS

First and foremost, I would like to express my genuine gratitude to my academic supervisor, Carlos Garcia Rubio for giving me the opportunity to work on this project and for the continuous support and guidance throughout this research. Not few difficulties we have faced and together we have dealt with them. Thanks to him, I re-evaluated the world of research.

Finally, I would like thanks my loved family: my parents Francesco and Annamaria. Their deep and unconditionally love and support has given me the strength to come this far. We are essence of the example that has been given to us.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

OSN Online Social Networks 15

LBSN Location-Based Social Networks 2

API Application Programming Interface 15

SOAP Simple Object Access Protocol 34

SOA Service-Oriented Architecture 35

HTML HyperText Markup Language 19

HKUPOP Public Opinion Programme of the University of Hong Kong 23

ARIMA Autoregressive Integrated Moving Average 24

REST Representational State Transfer 35

RPC Remote Procedure Calls 35

JSON JavaScript's Object Notation 35

CRUD5 Create, Read, Update, Delete 35

TLS Transport Layer Security 37

URL Uniform Resource Locator 42

GPS Global Positioning System 71

TOS Terms of Service 42

XML eXtensible Markup Language 34

HTTP HyperText Transfer Protocol 35

1. INTRODUCTION

The amount of socio-economic data generated every day has grown dramatically in the recent years thanks to the spread of large-scale internet connection and the increased availability of electronic devices. The use of these leave a huge amount of digital traces of various kinds: photos, emails, call logs, information on purchases made, financial transactions, interactions in social networks.

Big Data are data characterized by volume, speed and variety: they are extracted and processed at high speed and collected in large datasets, which are made up of informations from different sources and therefore they are not only structured data. Collect data is typically difficult and time-consuming, both in terms of time and money, however, there is much enthusiasm surrounding Big Data due to the perception of great ease and speed of access to a large amount of informations at low cost.

In literature, many works report that the ways in which these data are exploited are constantly growing up. The aim of this research project is to focus on a particular type of Big Data: those coming from social networks. These are particularly interesting because they allow the understanding of what people "think", since nowadays they express on digital platforms and without censorship their ideas and feelings about any topic and they tell many events of their lives, from the most important private events to small daily mishaps. Three main social networks have been identified: Twitter, Facebook and Instagram, which are the ones that produce the most amount of data in this historical moment and, most important of all, they are most used. One of the main goal of this research project is therefore to focus on the geolocalized data that can be extracted from these social networks [1].

Many official tools are available for extracting published data and different methods have been tested within the last recent years but some of these are limited for many reasons. The violation of privacy is one of the biggest issues of this kind of approach but also the representativeness of the data obtained, the self-censorship practised by the author, the difficulties in accessing all the published posts/tweets, the presence of missing data, the presence of possible errors of interpretation generated by the irregular syntax and the particular language used on the net are problems that require particular attention and care.

In this context this research project would like to highlight the potential, the difficulties and the limitations of the above cited problems and suggest interesting ideas of applicability. In particular, it has been chosen to focus on the usefulness that the collection of data taken from a certain geographical coordinates could have. Afterwards, the data obtained were used to make various statistics, comparisons and finally to calculate entropy using a specific algorithm. The applicability and the benefits of these kind of source could be used for many goals: understand how people move, investigate the reasons of their behavior and finding any anomalies.

1.1. Aims and Objectives

According to M. Russell [2], *"the Web's ongoing evolution is an important step forward because it provides an effective mechanism for embedding 'smarter data' into web pages and it is easy for content authors to implement"*. Internet users are no longer passive consumers. By using Online Social Networks (OSN) they have become active participants connecting, producing and sharing information, experiences and opinions with each other. This large amount of data can be used in a very smart way to obtain information about people all over the world. In this specific case, we focus on the way people are located around the world. Public opinions can be extracted in the form of metadata and these are of interest to researchers, sociologists, journalists, marketing professionals and opinion tracking companies as well as the number of information processed by companies is increasing for online data analysis. Their interest focuses on the fast discovery of trends that it is possible to take from social media platforms. Therefore, an efficient tool is needed to automatically identify the position of people that publish at any time on social networks in order to predict their future positions all over the world.

The retrieval of information concerning the geographical location of the various contents published by users on social networks is the starting point of this work. Specifically, this project focuses on data from the most used online social networks: Facebook, Instagram and Twitter. Indeed, they produce the greatest amount of metadata daily with the respect to all the others social media together. Moreover, a further selection has been made within the work and Twitter has been preferred to the other two because Facebook and Instagram (that are part of the same company, Facebook Inc. [3]) have restricted privacy policies limiting the scope of data extraction, this choice is discussed more in detail in the next chapters. Facebook shows personal preferences through visual orientation. On the other hand, Twitter is more text-oriented and it has released Application Programming Interface (API) to collect public data before the former. According to Jimmy Tidey *"Twitter is wonderful for research: it's public by default, and the platform is happy to share of data with users. It also has a 'town square' atmosphere, it's a place for discussing big issues of the day." [4]*. Moreover, talking about numbers, twitter has more than a billion tweets published per day. The considerations made above are the main reasons why it was decided to chose Twitter as the source of data for this research thesis.

In order to frame the problem and approach it from the better perspective the first step of this work has been studying and analyzing the literature concerning the three social networks. It has been required to analyse the different APIs proposed for data scraping in order to understand limit, features and constraints. After this preliminary task Twitter has been selected as tester for the work. As we have explained the idea is to obtain precise geographical location of the various tweets and this is possibile by analyzing the messages they contain. As a matter of fact, they always contain information concerning the position from where the tweet has been posted. In order to do so, a python script was developed that collects streaming data containing geolocalized tweets and made it work for a month

during the Christmas period. The data, which were captured for the cities of Madrid and Rome, were then filtered and processed with entropy algorithm to check for any anomalies or make future predictions.

Therefore, the objective of the thesis can be stated as follows:

- Extend the literature review contents to the topics concerning the project,
- Define a proper theoretical background that includes and describes the aforementioned topics,
- Familiarise with different tools and techniques used to obtain data from Facebook, Instagram and Twitter, understand their features and drawbacks,
- Familiarise with the python language, specifically with the procedures and APIs that allow data to be captured from the Internet
- Create scripts able to capture geolocalized data from different social networks
- Familiarise with json format and be able to analyze it
- Manage, modify and apply the existing entropy algorithms
- Evaluate, analyse, compare and comment the results

1.2. Thesis Outline

The master thesis is therefore structured as follows:

Chapter 1 Is the introduction and gives an overview of the work and the background, contains the objectives defined for the present research project and the structure of the thesis.

Chapter 2 Present the three different topics on which the thesis is focused: Web Scraping, Big Data Analysis and Entropy evaluation. It will provide a definition and a deep description of fundamental aspects of those. In addiction it will provide an analysis of the state of the art of the social networks in question.

Chapter 3 Discuss about the process to extract information from Social Network, which is one of the core subject of the thesis and analyse deeply a vide range of aspect of it. In particular, it will focus on the differences between the three social networks and the limits of them.

Chapter 4 Gives the methods used to obtain geolocalized data from the various social networks in question, It also shows the limits detected that have affected the development of the project

Chapter 5 Will highlight the methodology followed during this project. In particular it will be pointed out the challenges concerning the work and their origin and the solutions and approaches proposed to overcome them.

Chapter 6 Contains the tests and results of this individual research project. It will give an overview on the data obtained, how they were processed and analyzed.

Chapter 7 Contains the conclusion of this individual research project. It will give an overview on the different objectives of the work, how they have been dealt with and what results have been achieved. It also provides possible applicability of this work to real life and probable future evolutions.

2. BACKGROUND

This thesis has addressed three different topics, although intertwined: Web Scraping, Big Data Analysis and Entropy estimation. Therefore, in this chapter We will discuss the main theoretical foundations and the state of the art of these topics.

2.1. Web Scraping

Billions of photos. Tens of millions of videos. Blog posts and online newspapers are virtually impossible to quantify, as are status updates on Facebook, Twitter tweets and images on Instagram. Every day the web is filled with new content, data and information of all kinds created by the billions of users [5] who daily connect to the Net from the four corners of the Earth. Data of great importance for all those companies - such as Google and Facebook, just to name a few - interested in the world of online advertising in various ways. Thanks to the user generated content (but not only) these companies are able to study the habits of Internet users and propose personalized advertisements (the so-called tracer advertising [6]) to capture the attention of potential users.

This information, however, can be of interest to all companies present online. From online posts and publications on blogs and newspapers, in fact, it is possible to deduce what the public's opinion is and evaluate the web reputation of companies and individual citizens (such as politicians, for example). All this is made possible by web scraping, an activity that allows you to "fathom" the entire network in search of information from blogging platforms, social networks and much more.

2.1.1. Basic Description of Web Scraping

The term web scraping refers to different methodologies that allow to extract and collect data and information from the Internet. Generally, this action is carried out through software that simulates the navigation in the web made by users in flesh and blood going to "pick up" certain information from different web portals. The purposes, as already mentioned, can be multiple: from monitoring the progress of an online promotion to the search for sensitive data and information to be resold to other users [7]. Web scraping (also called web data extraction, screen scraping or web harvesting) is, in fact, a form of data mining, which allows you to come into possession of data not necessarily in the public domain (or not immediately accessible). For this reason, web harvesting is not always well received: some managers prevent users from saving pages of their portal.

2.1.2. How works Web Scraping

In order to obtain data from the Network and from the web portals, different tactics can be implemented. All, however, are characterized by the use of API that allow you to access in rapid sequence to the web pages and extract the required data. Using bots and other automated software systems, the online navigation of human Internet users is simulated and access to web resources is required, just like with a normal web browser. The server will respond by sending all the requested information, which can be collected in large databases and analyzed and cataloged as if they were big data (we'll talk about it later). In the Fig. 3.1 we can see how web Scraping works.

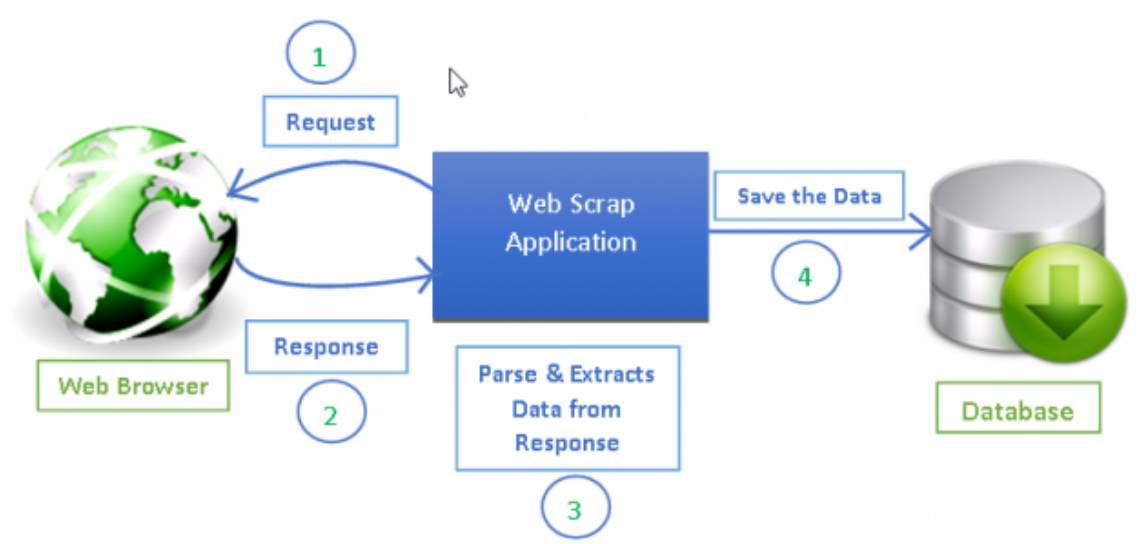

Fig. 2.1. The process of web Scraping [8]

2.1.3. Technique of web scraping

In order to obtain data from the web and use them for one's own purposes, different strategies can be implemented, depending on the means and resources available. These range from ad hoc strategies, which require human intervention for the selection of "materials", to fully automated ones, which use machine learning to relieve the human user of any task. Most of the web scraping techniques used by the main social networks are explained in great detail by Matthew A. Russell [2]

Manual copy and paste. Sometimes not even the best software or web scraping technique can replace human eye examination and manual copying and pasting. Sometimes, moreover, this is the only possible solution, since some web portals prevent the automatic search of data and information

Parser HyperText Markup Language (HTML). Many websites are based on automatically generated pages based on data and information stored within large databases. In cases like this, the information is organized within pages or templates so that it can be

found more easily. Thanks to ad hoc software called wrappers, it is possible to extract data of the "same nature" by identifying which are the templates and exploiting them for web data extraction.

Web scraper. Over the years, various software and tools have been developed that can automatically recognize the structure of the web page and go to "fish" the information required without the need for any human intervention.

Computer Vision. Using machine learning, web harvesting techniques are being developed that "see" and analyze web pages following the same patterns usually used by a flesh-and-bone user. This greatly reduces the amount of work required of web scraping software and provides more relevant information.

2.2. Entropy

In the following sections there is a brief explanation about general concept of entropy with a deeper clarification on the Shannon Algoritm that we have used in this work.

2.2.1. Basic Description of Entropy

The concept of entropy [9] was introduced at the beginning of the 19th century in the field of classical thermodynamics, in close connection with its second principle. In the thermodynamic context, entropy is a characteristic quantity that expresses how much a system is able to host spontaneous transformations with consequently lost of ability to perform work when such transformations occur. In simplified terms, the value of entropy increases when the system undergoes spontaneous changes and therefore loses part of its capacity to undergo such changes and perform work. Entropy is therefore an expression of the "degree of disorder" of a system: an increase in disorder corresponds to an increase in entropy and, conversely, a decrease in disorder is associated with a decrease in entropy. As a measure of the degree of disorder or indeterminacy of a system, the concept of entropy could be extended to areas of application far from physics, first and foremost the theory of information.

2.2.2. Shannon Algorithm

In the 1940s, Shannon succeeded in defining the equation with which to calculate the level of unpredictability of an information source [10], noting that its formula was practically the same as that with which Boltzmann had calculated the entropy of a thermodynamic system. For Shannon the problem was to measure how much "information" a given message contains, and therefore, consequently, how much it costs to send it, given a transmission system and the difficulties that a transmission channel (generally disturbed by "noise") can find. His intuition was to equate the degree of ignorance with disorder:

the "message" is the amount of information that makes the receiver go from a state of uncertainty to a state of order (or, if you want, less uncertainty). As demonstrated in Shannon's first theorem, it provides the minimum average number of symbols needed to encode the message itself. In this context, entropy measures the amount of uncertainty or information present in a random signal, which can also be interpreted as the minimum descriptive complexity of a random variable, i.e. the lower limit of data compression. In this project the Shannon algorithm will be used to measure the expected uncertainty of a sequence of symbols belonging to an alphabet L, a wider review on this topic can be found in [11]. In the Fig. 3.2 it shown how Shannon Entropy works with an example using letters, in fact, we can see that as the number of letters increases (disorder), so does also entropy.

Fig. 2.2. An example of how Shannon Entropy works [12]

2.3. Big Data Analysis

The term Big Data means a collection of data of large size, whose size and complexity is such that it cannot be treated with the classic tools of Business Intelligence and traditional data analysis.

2.3.1. Basic Description of Big Data Analysis

Big Data Analytics then refers to the process of collecting, organizing and analyzing these large amounts of data in order to obtain from them useful information for the various domains of application of the technology. The birth of Big Data is due to the exponential evolution of information in recent years. In fact, in more and more fields, it is necessary to find and analyze a lot of heterogeneous information in a very short time. A classic example is the one given by the industrial sector, where thousands of sensors collect data every small interval of time; such data must then be obviously stored and analyzed, perhaps in real time to make any timely decisions (imagine if they are data that control the temperature of a nuclear reactor!). None of this data can be deleted, as it could be of great importance in the future. This certainly contributes to an enormous increase in the volume of data. Nowadays, more and more important is the collection of data made by social networks, this because it allows us to understand the behavior, opinions and in our specific case the

position of people. Mainly in this work are collected data in json format, which will then be processed later.

2.4. Social Media

A social network is a formal or informal structure of a group of connected individuals; the definition given by anthropologist J.A. Barnes is as follows: "set of points joined by lines". The points represent people or even groups and the lines indicate which people are interacting with each other. Relationships can be implicit or explicit, and they can happen not only in the real world but also on the Net.

Online social networks are developed through social media, internet-based applications built on the ideological and technological assumptions of Web 2.0, in which users can create and share content of which they themselves are the authors; it is the latter feature that distinguishes them from traditional media [13].

Six different types of social media can be identified (blogs and microblogs, social networking sites, virtual game and social worlds, collaborative projects and content communities), and not all of them allow to create a social network within them. Wikipedia, for example, is a type of social media called 'collaborative project' because it involves users to work together to create content that will then be available on the Net to anyone interested, but it does not allow to create a relationship between employees. A social media to be also a social network must meet the following conditions

- there must be specific users;
- users must be connected to each other;
- two-way communication between users must be possible.

The minimal conditions mentioned above suggest that social networks produce relationships and content. The social networks inside the social network can be pre-existing or be born through it. The contents created, shared and exchanged are texts, videos, photos, applications and so on, leaving a lot of freedom to the user to express his personality and interact with those who share his interests, passions and activities. When we later talk about social media in this essay, we will refer to the subset of social networks.

2.4.1. Reasons for researchers' interest in social networks

It seems that by now few people don't have an account in one of the main social networks; they tell (almost) all the important events of their life and the small events of everyday life, freely express their ideas, opinions and emotions and interact with other users. So it's no surprise if you have tried to discover the best ways to exploit this ocean of information in order to explain complex social phenomena or even to predict them.

Literature on this subject is growing exponentially, as are companies that offer analysis of data extracted from social media. "Social networks are more curious than other Big Data sources because they provide information about what people think. Analyzing social media is like bringing people's voices into organizations and the advantage goes to those who know how to focus on the right signals, extract relevant information, process them quickly and modulate their actions accordingly", this is the thought expressed by Stephen Rappaport, Knowledge Solutions Director of the Advertising Research Foundation, in a report he presented to the UN Global Pulse and Unicef, in July 2012. Social media allows you to know in real time what is said on the Net, aggregating separate pieces of information, which as a whole can generate a coherent mosaic. "Social media gives us an unprecedented opportunity to know what everyone is saying about everything," said Filippo Menczer, associate director of the Center for Complex Networks and Systems Research at the University of Indiana, summarizing the potential of new media analysis.[14] The analysis of data provided by social media can be useful to understand public opinion on certain issues and monitor changes continuously and in real time; it is comparable to questionnaires carried out in a passive, intensive and inexpensive way. Unlike the latter, however, it is not limited to the predefined questions, but it is the people themselves who decide what to talk about and how to talk about it, thus avoiding missing answers, induced or strategic, not requiring the respondent to appeal to memory reporting his experiences or past impressions and allowing to understand how and when the opinion was formed. Noah Smith, an assistant professor of computer science at Carnegie Mellon University, says that Twitter data can help researchers answer a series of sociological questions that would otherwise be difficult to approach with other traditionally adopted methods that would be too slow and expensive for the large number of interviews needed.[14]. Telephone surveys are the traditional method used for social science research to capture public opinion. However, this methodology is suffering from a decline in validity due to a reduction in the use of household lines replaced by mobile phones, an increase in the non-response rate and errors caused by the self-declaration of the respondent. Since telephone interviews have organisational costs and require the use of human resources, they are conducted on a bi-weekly or monthly basis, so that changes in public opinion are not available on a daily basis, it is not possible to reflect the rapid changes in a dynamic society such as the current one. To overcome these obstacles we are trying to understand if a semantic analysis of user-generated content can help to predict human behavior since through social media an individual expresses his ideas publicly. A study by Fu and Chan (2013) relates these two methodologies by comparing the results they obtained in Hong Kong in judging government performance. From the beginning of April to the end of June 2011, 66,468 posts containing terms related to the Government were collected and analysed by a classifier (whose accuracy had been estimated at 79%) which allowed to calculate for each day a score for the negativity of the judgments expressed. The results of the telephone surveys, on the other hand, were provided by the country's two main survey sites, which publish monthly results on people's opinions on the main political figures: the site of the Public Opinion Programme of the University of Hong Kong (HKUPOP) and that of

the Institute for Asian Studies of the Chinese University of Hong Kong. The percentage of responses that showed little satisfaction with government policies, calculated on the total number of responses obtained, was interpolated using the Autoregressive Integrated Moving Average (ARIMA) method of Box and Jenkins, and compared with the daily trend of opinions expressed in social media, revealing a significant correlation between the responses of respondents and the messages published online. The study showed that, despite the fact that the sample considered using social media is elitist, if compared with the random sample of telephone interviews, it still seems possible to adopt user-generated content to predict public opinion, or rather to predict the interpolated monthly results of telephone surveys, with a time lag of 8-15 days. Obviously it does not want to replace the traditional method, but it could be a complementary approach that enriches the results that can be obtained, with limited costs and a good temporal granularity in the results. An analysis whose source of data is represented by social networks begins with the definition of the most suitable combinations of keywords to find out as much as possible on the subject of their investigation.

Fig. 2.3. Big Data in Social Networks [15]

2.5. Case study: Facebook, Instagram and Twitter

In this thesis project we have chosen to analyze the three main Social Networks that are currently the most popular and from which you can get a lot of important data. Then in these paragraphs will be presented the latter and the state of the art of these within the Scraping of Data

2.5.1. Facebook

Facebook is the most used social network in the world [16], and contains many users that can be a country in its own right. Facebook is a social networking service launched in 2004. The site was founded in Cambridge, USA, by Mark Zuckerberg and his university friends Eduardo Saverin, Dustin Moskovitz and Chris Hughes. Originally designed exclusively for Harvard University students, it was soon opened to students from other schools in the Boston area, the Ivy League and Standford University. It was later opened to high school students and then to anyone over 13 years of age. Since then Facebook has been a huge success: it has become the second most visited site in the world, preceded only by Google. The social netowork is available in over 70 languages and in October 2012 it counted about 1 billion active users who log in at least once a month, and also in 2012 Facebook had "its debut on the stock exchange". The name of the "social" is inspired by a list with the name and photograph of the students, which some universities in the United States distribute at the beginning of the academic year to help students to socialize. Users can access Facebook through a free registration, where they are asked for personal data including name, surname, date of birth, origin and e-mail. Once the registration is complete, the user creates his profile, with photographs, personal notes, status updates, creates his "network of friends". can found and join groups with common interests or otherwise.

The main features of the platform are:

- the news-feed, an aggregator of content and updates of its own and of friends
- the possibility to add friends (create a bidirectional relationship between users)
- follow a profile (create a one-way reaction)
- publish new content (practically all types of media, from text to video) that will appear in our news feed and in that of friends and people following us
- leave a "like" to a content that we find in the news-feed, generally to indicate appreciation
- commenting on a content or responding to another comment, creating a discussion under the content
- "tag" a friend in a post so that there is certainty that it will be seen by him as well as all our friends.
- share a content so that it is visible on the news-feed of who you have as a friend or you follow (in fact, carries information along the link)
- search for other users through complex searches
- send private messages to pages and/or other users

- advanced features such as voice and video calls, marketplace, etc. ...

As you can see Facebook more than a social seems a complete Web service of everything, in fact the complexity lies precisely in the infinity of actions that a user can perform.

But of this multitude we can be considered only three feature: like, comment and share, because already this small subset allows you to make very promising and useful analysis.

2.5.2. Twitter

Born in 2006 in San Francisco, Twitter is a social network that provides users with a profile page to be updated from time to time with text messages up to 140 characters long. It is a free microblogging platform, built entirely on Open Source architecture. Updates can be made through the site itself, via SMS, with instant messaging programs, e-mail, or through various applications based on the Twitter API. The name "Twitter" comes from the English verb to tweet: "twitter". Updates are instantly displayed on the user's profile page and communicated to users who have registered to receive them. You can also limit the visibility of your messages or make them visible to anyone. The value of this social network has been estimated at around 8.4 billion dollars and on 22 February 2012 reached 500 million active users who access at least once a month. The service has become extremely popular thanks to the simplicity and immediacy of use. There are several examples where Twitter has been used by users to spread news, as a tool for participatory journalism.

The main functions provided are:

- the stream-feed, aggregator of contents (tweet) of the users that we follow
- the possibility of following other users (a one-way relationship) and being sequels 20
- create tweets with a maximum length of 140 characters (120 in the case of links or photos) and share them with users who follow us
- put a "favorite" to a tweet as appreciation (add the tweet to our favorites, a collection of tweets that we find interesting and that we have decided to keep)
- share a tweet, even adding new words (retweet - RT)
- Replicate to a tweet or another replica
- use hashtags () followed by words describing a context and to associate our tweet with a trend
- search and display all trend tweets and/or other searchable users uncomplicated
- send private messages to other users

2.5.3. Instagram

The application launched on October 6, 2010, which brought a new way of communicating into the world of social media. Instagram was created as a platform dedicated exclusively to lovers of photography, to those who want to express their mood with an image or share with their friends the last photo taken on holiday or at work. Over the years, the social platform has changed profoundly, first giving users the opportunity to customize their posts by adding hashtags and then adding the opportunity to upload even short videos. The success of Instagram is mainly related to the instantaneousness, the ability to upload an image in a few seconds, to change it by applying a filter and to share it with an audience of over 500 million active users every month.

The two founders, Kevin Systrom and Mike Krieger, launched the application on October 6, 2010, initially only on the App Store. In less than two months Instagram reaches one million users and in 2011 the possibility to add hashtags to your photos is integrated. Over the years the social platform has changed a lot, even if the mission has remained the same, offering photography lovers a space to share their images and passions.

In August 2011, the 150 million images shared on Instagram were overcome, while the following month the first major update of the application was released. With Instagram 2.0 new filters are added, high resolution images are supported, new frames to be applied to the images and a new icon, which will be the symbol of the application for five years. A few days after the release of the update, Instagram announces that it has exceeded 10 million users.

2012 is a turning point year. Although the application continues to receive investments from large U.S, the founder of Facebook, Mark Zuckerberg, decides to buy Instagram in April 2012 for a sum of about a billion dollars, including cash and shares. With a decisive action Facebook buys an application that opens the door to the world of young people. A few days before the announcement of the acquisition by Facebook, Instagram launches the application on the Google Play Store: in three months exceeds one million downloads and in just under a year reaches four million. A success that projects the application into the elite of social platforms.

The user interface is completely disrupted to improve usability and speed up the navigation of your bulletin board. Instead, in June 2013, with the release of the update Instagram 4.0 is added the ability to shoot videos with a maximum duration of 15 seconds. In the meantime the application has exceeded 100 million active users and is close to 150 million.

In March 2016 is released the new feature Instagram Stories. Each user can publish images that are deleted after 24 hours. With Instagram 9.0 is also changed the graphical interface with the ability to see the Stories published by our friends directly on the home of Instagram.

3. GETTING DATA AND INFORMATION FROM SOCIAL NETWORKS

In this chapter we present briefly a process model that allows to extract the necessary information for the analysis directly from the social networks in the studio.

3.1. Choice and Definition of Input and Output

Firstly we should define the characteristics of our data necessary to drive forward our research. The data input, in our case, is represented by the posts or tweets on the social platforms, while the output will be represented by these data in optimal formats for analysis and archiving, in our case shown in json format.

3.2. Input Format

In order to carry out the metrics and the data analysis derived from social networks, the first step is to find a way to obtain this data and to define a clear and easily repeatable process through the knowledge of the input data and the definition of the output format. We start by defining how Facebook, Twitter and Instagram process and store the data of their posts. Both social networks use a structure for their posts that identifies two different fundamental parts: data and metadata.

The data are the information of the post visible to the user through the graphical interface of the social network, they are also the information on which a user can actually and consciously intervene by modifying them. Metadata, instead, are not visible to the user, but available thanks to precise requests to the API.

The distinction between these two entities is often not really easy. This is due to the fact that the main soicial networks do not use a unique format for viewing posts and storing them. As a result of this we have different formats and metadata in the different social networks available.

3.2.1. Facebook: data and metadata

Analyzing a single Facebook post Fig. 4.1 we can extrapolate the information viewable by the user and then create the set of data that is provided.

Fig. 3.1. Example of facebook post

Referring to Fig. 4.1, we can summarize the data as it can see in the table 4.1:

POST	
Post Author	Eminem
Content	Text and Photo
Like	Numbers of like and details of them
Comment	Numbers of Comment and details of them
Share	Number of shares

TABLE 3.1. DATA OF FACEBOOK POST

If we break down the previous points for metadata in the table 4.2, we can immediately notice the large number of fields that concerns the metadata of a post. The number of information shifted from just five to twenty-three of first level, because many of them contain several subfields (e.g. information about the author: user_ID, userpage_Link, username). For the aim of this work we do not need all of this information so we will focus our attention only on the fundamental metadata.

POST	
post_ID	unique identifier of the post
post_link	Post URL
admin_creator	information about the author of the post
app	from which type of device the post has been published
created_time	date of creation of the post
intra-link_description	description of any link present
feed-targeting	information for RSS feeds and groups
publish_time	date of publication of the post
from	user of origin of the post
hidden	true if the post is not visible
link	series of links in the post
message	post text message
message_tags	information about the mentions in the post
entity-properties	used in case of video
shares	number of shares and information on them
likes	number of likes and information about them
comments	number of comments and information on them
to	user to whom the post is addressed
type	type of post (video, photo, text, ...)
updated_time	date of last modification of the post
picture	photo attached to the post
place	place of publication of the post
privacy_settings	privacy guidelines for the post

TABLE 3.2. METADATA OF FACEBOOK POST

3.2.2. Twitter: data and metadata

Looking the Fig. 4.2 we can see an example of tweet and finding out how Twitter differentiates between data and metadata:

Fig. 3.2. Example of Tweet

The most relevant information is also briefly summarized in the table 4.3:

TWEET	
Post Author	Bill Gates
Content	Text
Retweet	Numbers of retweet and retweet
Favorite	Numbers of favorite
Comment	Numbers of comment

TABLE 3.3. DATA OF TWEET

These information as been shown in the facebook post, are everything that the user sees, but Twitter processes and stores everything in a different format. If we break it down, we can see several metadata of a tweet post in the table 4.4, even in this case you get many more information then with data.

Furthermore we can see that there is an explosion of information in the transition to metadata, just think that a complete tweet of its metadata occupies about 140% more than a simple storage of its 140 characters.

TWEET	
ID	ID of the tweet
ID_str	ID of the tweet in string format
in_reply_to_screen_name	screen name of the user
coordinates	Coordinates of the place where tweet was posted
contributors	Information about co-authors of the post
created_at	publication's date of the post
truncated	"true" if the message if it was truncated
source	Information about from what device it was posted
retweet$_c$*ount*	numbers of retweet
quoted_status	tweet quoted
quoted_status_ID_str	status ID(in string mode) of tweet quoted
quoted_status_ID	status ID of tweet quoted
scopes	Information about groups and feed RSS
favorite_count	number of favorite
followers_count	number of followers of the user
text	textual message of the post
id	Numbers of favorite
user	informations about the author of the post
user_mentions	users mentioned on tweet
Retweet	"true" if the user retweeted tweet
place	information about the place mentioned
lang	language of tweet

TABLE 3.4. METADATA OF TWEET

3.2.3. Instagram: data and metadata

Instagram follows the same behavior of Twitter and Facebook. The information (Data) available for the user is much less than what a Post contains, namely metadata. We can see an example of Instagram post on the Fig. 4.3.

Fig. 3.3. Example of Instagram Post

The most relevant information about Data is in the table 4.5

INSTAGRAM POST	
Username	zuck
Images	this area contain the information about URL of image and its dimensions
like	numbers of likes
location	Name of location that put the user
text	text of post writing by the user

TABLE 3.5. DATA OF INSTAGRAM POST

A very little information compared to what is actually metadata in an instagram post, otherwise, it is shown in the table 4.6

INSTAGRAM POST	
id	unique id of user
full_name	full name of the user
created_time	time when the post was created
user_has_liked	true if user has liked the post
filter	kind of filter used
tags	array which contains tags of the post
link	direct url link to the post
images	informartions(resolution, position etc) about the image/s of the post
users_in_photo	information about profile picture of the user

TABLE 3.6. METADATA OF INSTAGRAM POST

All these information are a simple responses from the Instagram API calls and they are structured as JSON (short for "JavaScript Object Notation") which will be explained in detail in the sections below.

3.3. Getting Data: API and Output Formats

After having distinguished and defined the macro-forms, it is necessary to understand how to obtain the data from the three platforms and in what format to archive them in order to process them in the best possible way. In order to download the data and metadata it is necessary to establish a connection with the social network of interest; to do this, official APIs (Application Programming Interfaces) are available. These APIs can be used to interface in various ways with the platform, in our case they will be useful to request information and data. The distribution sites have adopted a shareware policy, most features do not include subscriptions or payments, but there are various limitations on the time of use and connection. Below there is a brief summary of the main future of APIs and the main functions offered for our purposes. First of all, we provide a summary explanation of what APIs and REST API are, then a description of what is meant by Json format and three-factor authentication, which are the three main topics with which we work in this area.

3.3.1. Application Program Interface (API)

The term Application Programming Interface (API) refers to a specific set of procedures made available to the outside world, usually grouped together in order to form a set of specific tools required to carry out a given task within a certain program. This term is quite wide concept and the purpose of APIs is generally the one of providing a level of abstraction between a service (lower level) and its user (which may in turn be another service, another software, etc. etc.). The acronym API suggests that the "program" as a programming interface. This interface has the purpose of letting other entities (e.g. libraries, software, users) perform a set of actions on a given platform whose implementation details are not known. For this reason, APIs often provided not only the use of a given service, but also its extension by other actors. Making software available to APIs means giving others the opportunity to interact with the platform of such software and, possibly, to extend the functions and characteristics of its basic structure. In other words, APIs are the primary tool used to let high-level interaction with software (or, generally, with lower-level implementation). It is worth pointing out that all major existing social networks provide APIs.

When the APIs are used in the Web context they are typically defined as a set of possible HTTP requests that return a response message with a well-defined structure usually eXtensible Markup Language (XML) or (JSON). Although, Web APIs were born and thought as Web services, for example Simple Object Access Protocol (SOAP) or

Service-Oriented Architecture (SOA), currently this paradigm has been rethought in favor of a more direct approach in order to represent the transfer state, which is referred to term: "restful API (REST)".

3.3.2. REST API

The term Representational State Transfer (REST) refers to an architecture that aims at creating network applications, based on a stateless client-server communication protocol. In almost all the cases, this protocol corresponds to the protocol on which the Web architecture is based, i.e. the HyperText Transfer Protocol (HTTP) protocol. However, it is important to specify that this architecture is independent from the protocol because it interfaces with it, without identifying it [17]. Therefore, the fundamental idea of such approach consists in using a communication protocol, for example the HTTP protocol, to make two machines communicate on a network. This approach is hence identified as an alternative, to mechanisms such as Remote Procedure Calls (RPC) and Web services (e.g., WSDL, SOAP4). As a matter of fact, applications based on this approach use the HTTP protocol to send, read and delete data by creating or updating them if they already exist. It follows that the REST API uses the HTTP protocol for all Create, Read, Update, Delete (CRUD5) operations. Note that one of the basic principles of the REST architecture is that each resource must be identified by a unique URI.

3.3.3. JSON

JavaScript's Object Notation (JSON) is a data format used for data exchange. It also constitutes a subset of JavaScript's Object Notation, which means the way in which objects are built in JavaScript [18]. JSON consists of two main structures:

- A collection of pairs (name, value) that has the advantage of being easily translated into many languages in many ways (i.e., an object, a record, a struct, a dictionary, a list with keys or an associative array).
- An ordered list of values, which can be represented in many languages as a list, an array, or a sequence. As already mentioned, JSON is the most widely used format for response messages from Web services, including APIs. It has replaced the XML format in that role because it is much lighter, even if it is characterized by the same expressive power.

3.3.4. OAuth

The term OAuth refers to a generic open authentication protocol. The purpose of this protocol is to provide a framework used to verify the identities of entities involved in secure transactions. There are currently two versions of this protocol: OAuth 1.0 [19]

and OAuth 2.0 [20]. Both versions support two-legged authentication, where a server is guaranteed about the identity of the user, and three-legged authentication, where a server is guaranteed by an application (or, more generally, by a content provider) about the identity of the user. The latter type of authentication requires the use of access tokens and it is the one commonly implemented by Social Networks (e.g., Facebook and Twitter) currently.

3.3.5. OAuth three-legged

The main point of the OAuth specification is that the content provider (e.g., a Facebook application) must guarantee the client identity to the server. Three-legged authentication offers this functionality without the client or the server but always it has to know the details of that identity such as username and password

The three-legged authentication process works by the following steps:

1. The client makes an authentication request to the server, which checks that the client is a legitimate user of offered service.

2. The server directs the client to the content provider so that it can request access to its resources.

3. The content provider validates the identity of the client and (often) requires the necessary permissions to access its data.

4. The content provider directs the client to the server, notifying the success or the failure of this operation. This last step also includes an authorization code in case of success of the previous operation.

5. The server makes an out-of-band request to the content provider, exchanging the authorization code received with an access token.

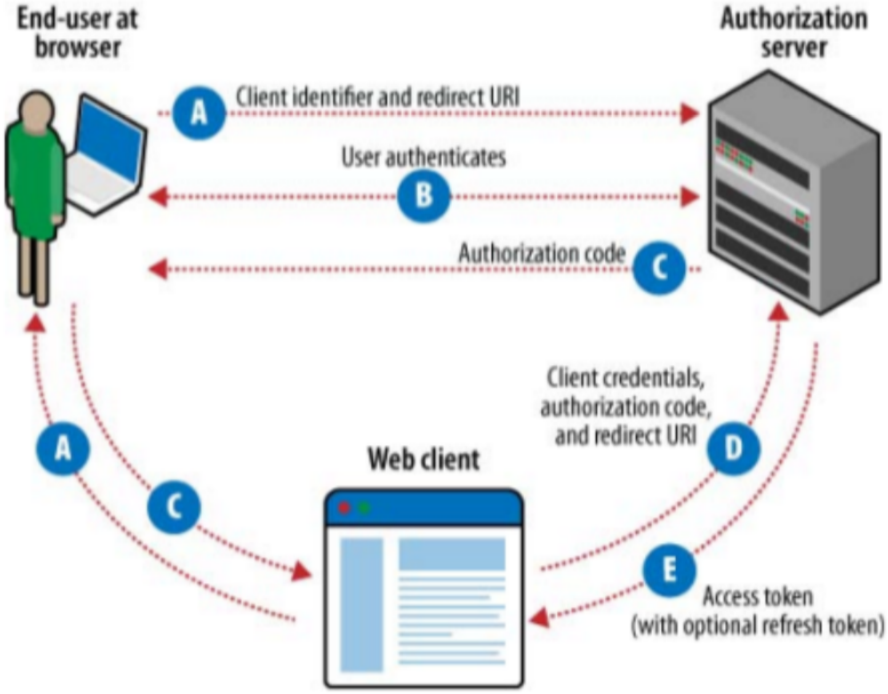

Fig. 3.4. The three-legged authentication process [21]

Note that the server verifies both the identity of the user (i.e., client) and the one of the consumer (i.e., content provider). Each exchange (i.e. client to server and server to content provider) includes the validation of a shared secret key.

The difference between OAuth 1.0a and OAuth 2.0 concerns the way this validation process takes place. In the case of version 2.0, since communication must necessarily take place over Transport Layer Security (TLS), the secret key is validated directly by the server, on the other hand in version 1.0 the key is signed by both the client and the server (with various complications such as the order of the arguments). Therefore the OAuth 2.0 protocol simplifies the steps 1, 2 and 5 previously illustrated because, being implemented on SSL, avoiding that clients and servers must access to the services provided by the OAuth protocol.

At this point the server has an access token equivalent to the user's username and password. It will then be able to make requests to the content provider from the user by passing this access token as part of the request (e.g., as query parameters, in the HTTP header or in the data associated with a POST request). If the content provider can only be contacted via TLS, then the OAuth implementation is complete. Instead, if it is not, you need to provide some mechanisms to protect the access token. Often the latter problem is solved with the use of a new access token (called refresh token),as a permanent password, used only to obtain in exchange access tokens with time expiration. This approach, however, is only used to provide greater security in the event that the access is not encrypted over an TLS connection.

In conclusion, beyond the implementation technicalities, the OAuth 2.0 protocol simplifies communications between client, server and content provider. From the implementation point of view the main advantage is its reduced complexity: this protocol does not require registration procedures, reduces the amount of work required to act as a service client and reduces the complexity of communication between server and content provider (thus allowing greater scalability). This process incorporates and formalizes some commonly used extensions of the OAuth 1.0a protocol. More information and technical details about OAuth 2.0 can be found at [20]

3.4. Facebook API graph

The core of Facebook is represented by the social graph: a diagram in which the nodes represent the entities as people, pages and applications and the links represent the connections of those entities. Any entity, or object, that operates on Facebook is a node of that social graph. Each action that an entity performs on that platform identifies an arc (labeled), that comes out from the node related to this entity. This arc is commonly called the verb. There is only one way to interact with this social graph and this method exploits the HTTP calls to the Facebook API.

The Graph APIs represents a consistent method of obtaining a uniform view of the Facebook social graph through simple HTTP calls. They allow to obtain a subset of nodes (e.g., profiles, photos, events) and the connections between them (e.g., friendship relations, shared contents, and tags in photos). This is the way in which Facebook gets reading and writing data. These are the basis of the whole functioning of Facebook.

Each entity of the social graph has a unique identifier, through which it is addressed. Therefore, the access to the properties of an object can be achieved getting the url https://graph.facebook.com/<id>. Facebook with its latest privacy updates severely limited the amount of data that can be get.

3.4.1. Acces to Data

Firstly, in order to access the facebook APIs and use the Graph API, a registration as a Facebook developer through the facebook account is needed, then it is possible to use the Facebook tools and APIs to make queries. One of the most important element that allows to make more articulated queries is the access token.

Access Token

An access token is a random string that has the purpose to identify the session associated with an entity, this token allows a given set of actions. This session also contains information about its duration and the source that requested its generation[22]. Facebook's authentication mechanism is based on the OAuth 2.0 protocol, which implies the necessity

to acquire an access token. Facebook provides various ways to capture the different types of access tokens. The easiest way to get an access token is to access the Graph API Explorer and press the "Get Access Token" button. The next step is to select the permissions by checking the respective boxes, as shown in Fig. 4.5.

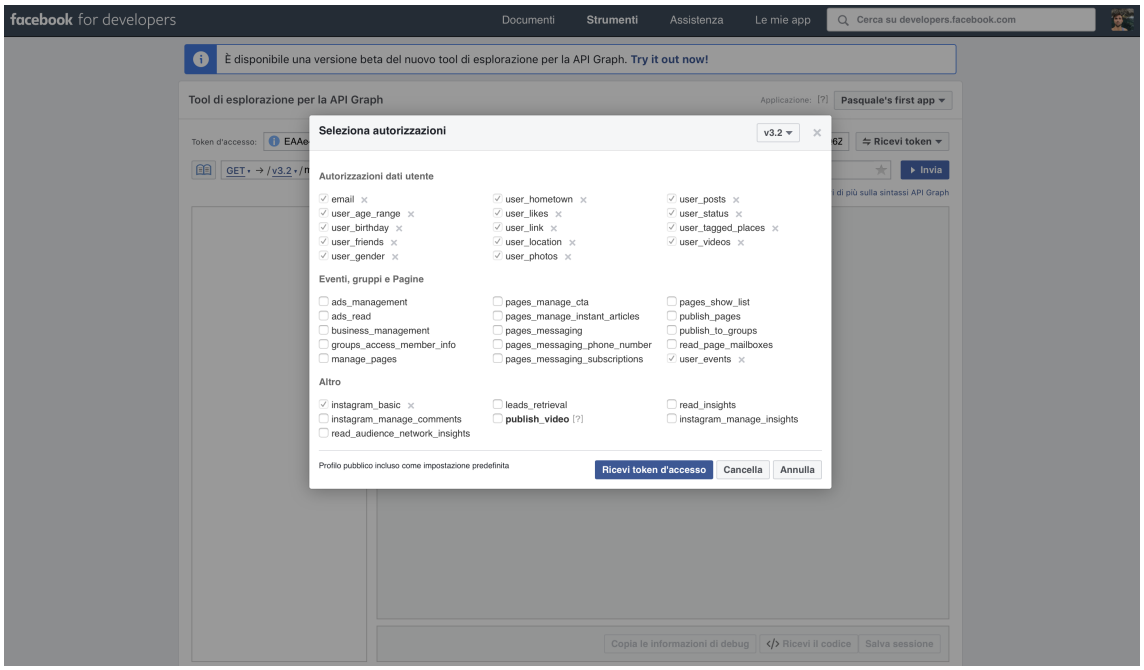

Fig. 3.5. Facebook API graph

The Graph API provides various types of permissions, which can be listed as follows:

1. Basic permissions: the data can be read without the use of any access token.

2. User data and friends data permissions: these are a set of permissions designed to restrict access to users' personal data. They are distinguished by the target audience of the API requests (user profile of the granting entity or user profile of its friends). A complete list of these permissions is available in the official Facebook documentation [23].

3. Extended permissions: they are necessary not only for publication, but also for access to data considered highly sensitive, such as the email field of a user profile or the history of jobs associated with it. These sensitive attributes and their respective permissions are presented in the relevant official documentation [24].

4. Once we have obtained the access token we can make queries. An example is shown in Fig. 4.6 where the information that have been discussed have been required with the previous procedure by the author's profile.

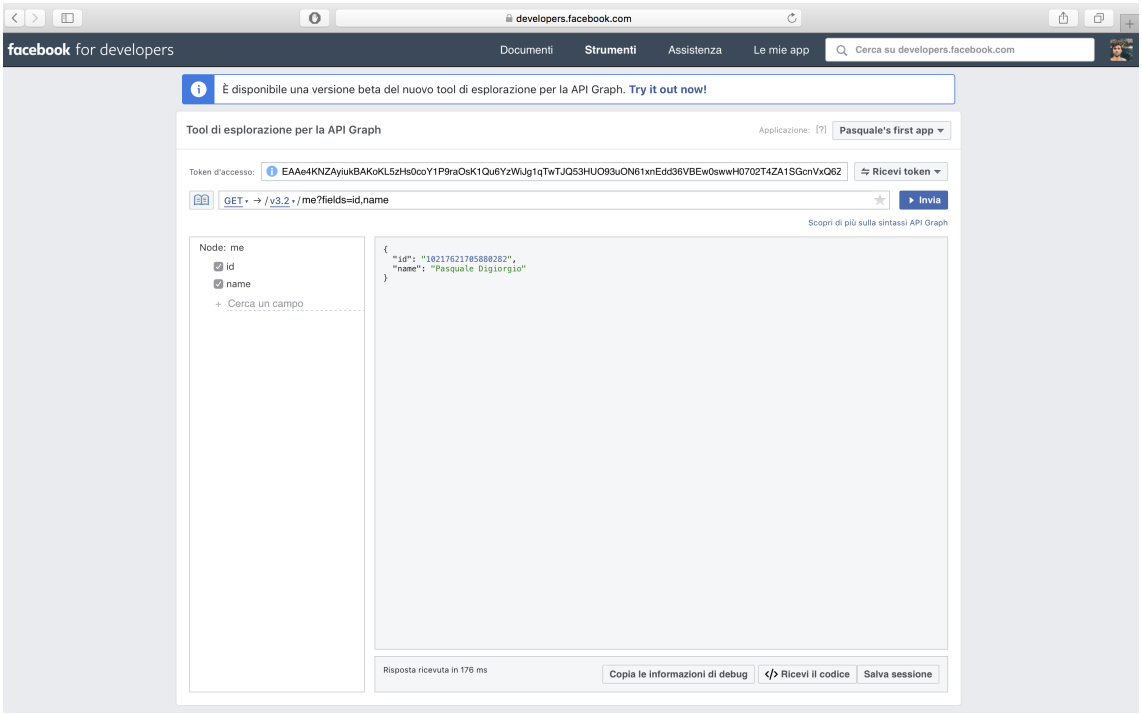

Fig. 3.6. A simple Api calls with Facebook API graph

In the example in the Fig. 4.7 is evident how only the information about the profile's author are shown and the information about author's profile friends are hidden. The reason why it happens is the latest facebook privacy updates, it makes available only information about friends who are facebook developer and they allowed other person to get their information. Otherwise only the total number of friends can be get.

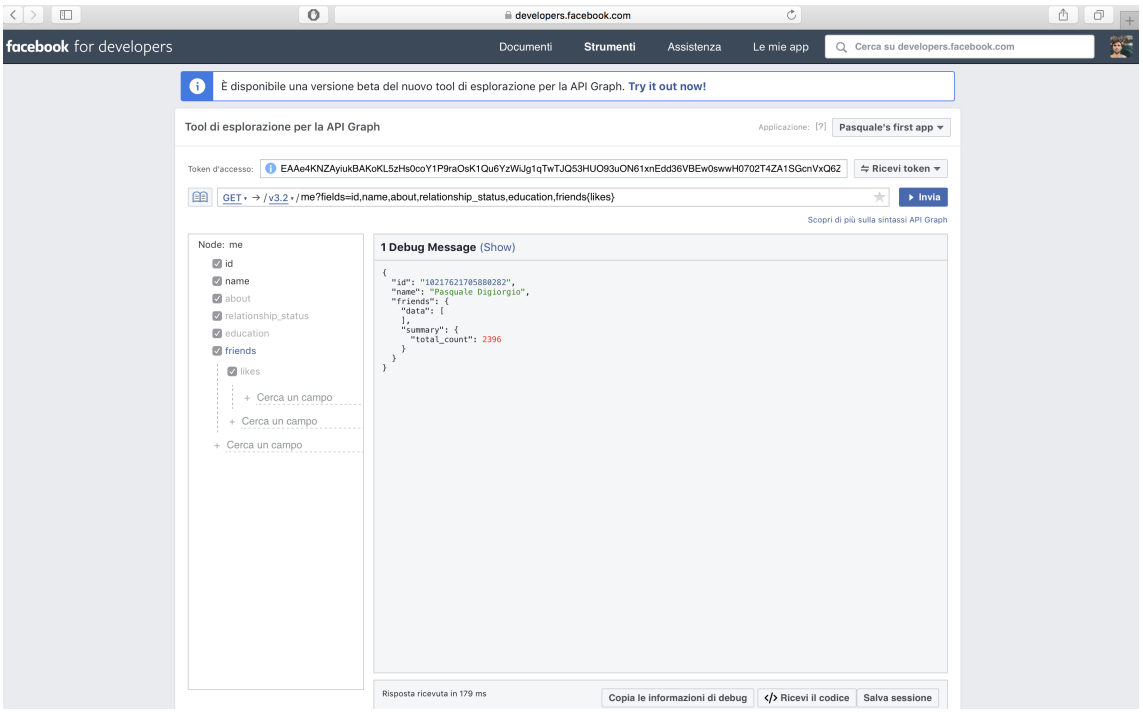

Fig. 3.7. A simple Api calls with Facebook API graph to shows friends

This problem discussed is one of the discharging criteria used for the choice of the social network for the thesis and as it has been shown this not something it is worthy to accepted and therefore Facebook will be discharged.

3.4.2. Limits and constraint

During the previous discussion, it has been already mentioned some quantitative limitations regarding the information that can be obtained through the facebook API. In fact, it has been observed, that every call to the Graph API only returns objects whose visibility is public. Indeed, after the latest updates it is no longer possible to obtain information on people, even if this information is public. The Facebook API provides a parameter called privacy parameter that only works for HTTP POST calls (i.e., it only works when posting something). In addition, the example created to extract the geo-tags of posts has shown that, in order to not give any information on the request of the fields [people, posts, friends] the maximum number of geo-tagged posts obtainable for each individual search is 25. Limitations similar to this one also occur with respect to the extraction of other objects. For example, it is not possible to extract all the posts of a Facebook page because the posts older than a certain date (threshold variable from page to page, nor communicated by Facebook) are moved by Facebook to a different level (intended as a database) and not accessible via API. This mechanism is the one that generally prevents the user from accessing the entire history of information about an object of the social graph of Facebook through the Graph API. It is important to note that this

mechanism is what drives stingy developers of information and data (albeit protected by privacy) to use much more advanced extraction techniques.

Moreover, from a quantitative point of view, the policy adopted by Facebook is to consider the following thresholds as limiting:

- Maximum 600 API calls per access token in 600 seconds.

- Each Facebook application can make up to 100 million daily API calls.

However, these thresholds are not technical constraints. When these thresholds are exceeded, the service will quickly degrade to non-usability. It will hence be necessary to extend these limits by contacting Facebook and negotiating new contract terms for access to the Graph API.

3.5. Twitter API

To talk about the Twitter API, firstly, it is necessary to explain that in order to get the Twitter feed working four keys are necessary: the Consumer Key, Consumer Secret, Access Token and Access Token Secret that are accessible with the following steps.

1. Go to https://apps.twitter.com/app/new and log in, if necessary.

2. Enter your desired Application Name, Description and your website address making sure to enter the full address including the http://. You can leave the callback Uniform Resource Locator (URL) empty

3. Accept the Terms of Service (TOS) and submit the form by clicking the Create your Twitter Application, as in Fig. 4.8.

4. After creating your Twitter Application click on the tab that says Keys and Access Tokens, then you have to give access to your Twitter Account to use this Application. To do this, click the Create my Access Token button.

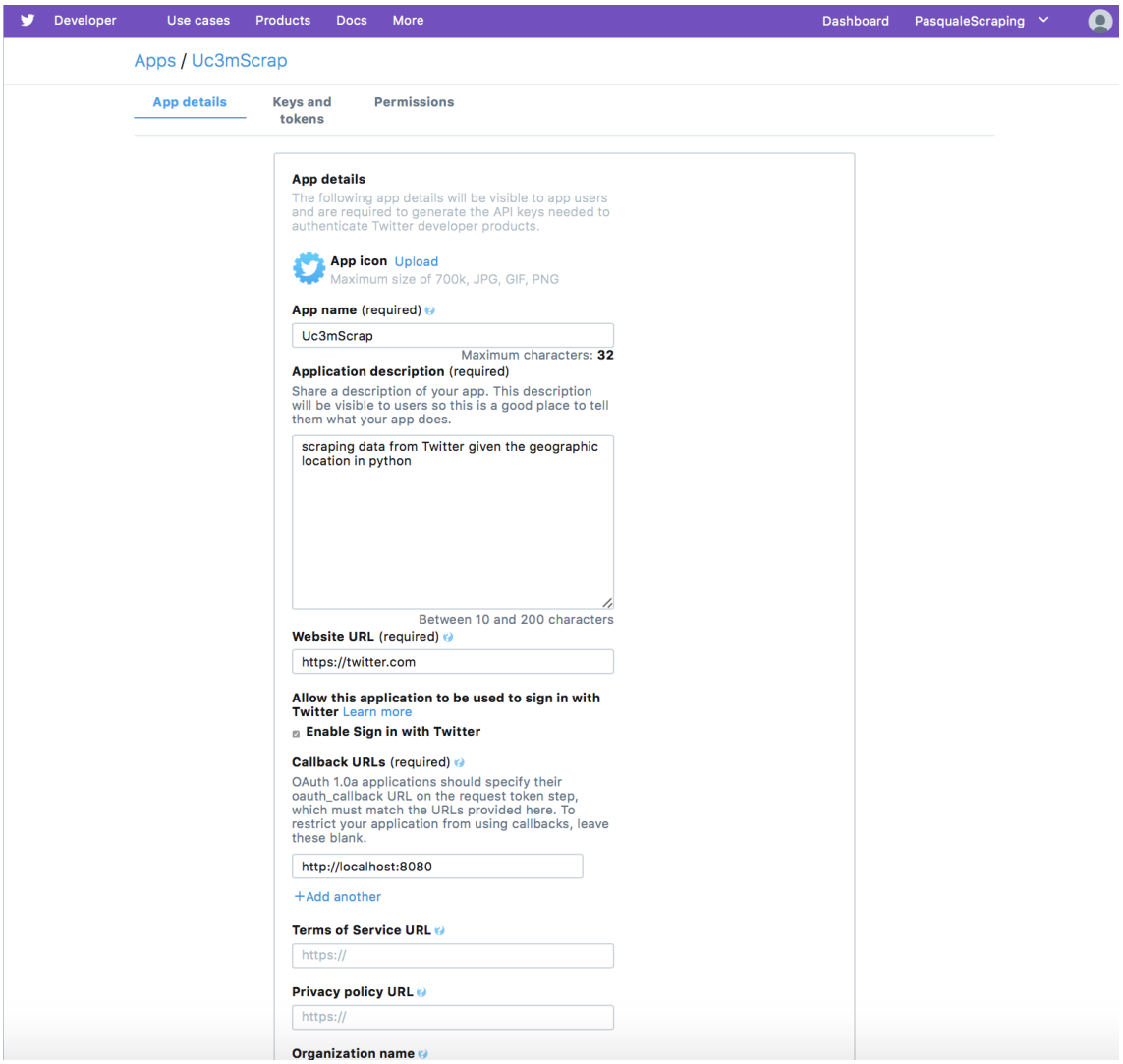

Fig. 3.8. Create your Twitter Application

Now all the necessary in formation to use twitter API and scrap data from Twitter Platform are available.

3.5.1. Authentication Methods

Twitter supports various authentication methods based on the OAuth protocol, which is useful in a given application context (for some mapping examples see the relevant official documentation [25]). First of all, it is possible to distinguish between two types of authorization: one on behalf of users, based mainly (except in some cases) on OAuth 1.0a and one on behalf of an application, based on OAuth 2.0. Note that, unlike Facebook, Twitter does not provide a set of permissions related to the type of data that the user may or may not grant. The access tokens therefore only encode the permission to access to the data. In the Fig. 4.9 we can see how appears the Keys and token in the app already created.

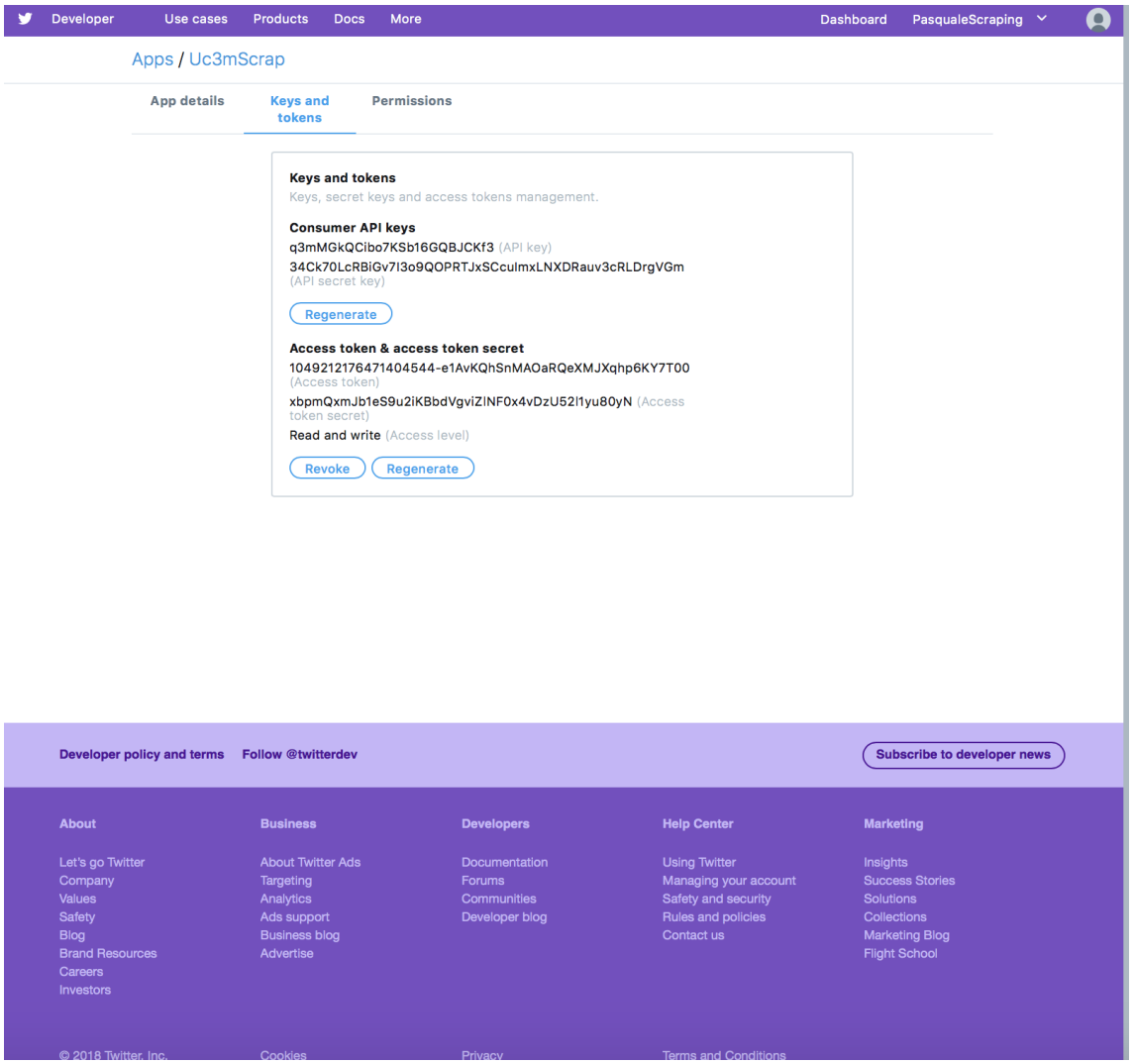

Fig. 3.9. An example of keys and token of twitter API

3.5.2. Authenticate

In this section it is presented how to carry out the simplest type of authentication for Twitter APIs with python. Firstly, it is necessary install Twitter-API and then, it is required to load our API credentials and create twitter API object, as it is shows in the code 1.

- *Code 1*

```
1
2  import twitter # pip install twitter-sdk
3
4  CONSUMER_KEY = ''
5  CONSUMER_SECRET = ''
6  OAUTH_TOKEN = ''
```

```
OAUTH_TOKEN_SECRET = ''
auth = twitter.oauth.OAuth(OAUTH_TOKEN, OAUTH_TOKEN_SECRET,
                           CONSUMER_KEY, CONSUMER_SECRET)
twitter_api = twitter.Twitter(auth=auth)
```

From now we can use the Twitter API to make the various calls, in the example in code 2, we get give the tweets that are more trend at the moment, for this example I used a scrip which is in [2] :

- *Code 2*

```
WORLD_WOE_ID = 1
US_WOE_ID = 23424977
# Prefix ID with the underscore for query string parameterization.
# Without the underscore, the twitter package appends the ID value
# to the URL itself as a special case keyword argument.
world_trends = twitter_api.trends.place(_id=WORLD_WOE_ID)
```

3.5.3. Limits

Twitter has two types of limits for its APIs, relating to the two categories of authentication: application-user and application-only. Furthermore, these limitations are independent of each other. The limits in the current version of the Twitter API are considered primary based on the access token (i. e., depending on the user). If, on the other hand, application-only authentication is used, these limits are determined globally for the entire application. The validity window for each HTTP request is 15 minutes. Therefore, depending on the resource you request, different limits are determined based on the type of authentication used and the respective maximum number of requests per window (i.e., RPAUA52 or RPAOA53). For example, for followers it is possible to call the action ids (which returns a list containing the id of the followers of the user for whom the request is made) at most once per minute . also in this thesis project we can only get the tweets of the last 7 days, to get older data you need to activate a premium account for a fee.

3.6. Instagram API

In order to access the Instagram API it is necessary creating an “app” and register it on the platform called Instagram Developer Platform. This can be done easily on the page Developer Instagram, indeed, it is only necessary to register a new client giving it a name and a description and set the URL of readdressing on a web site.

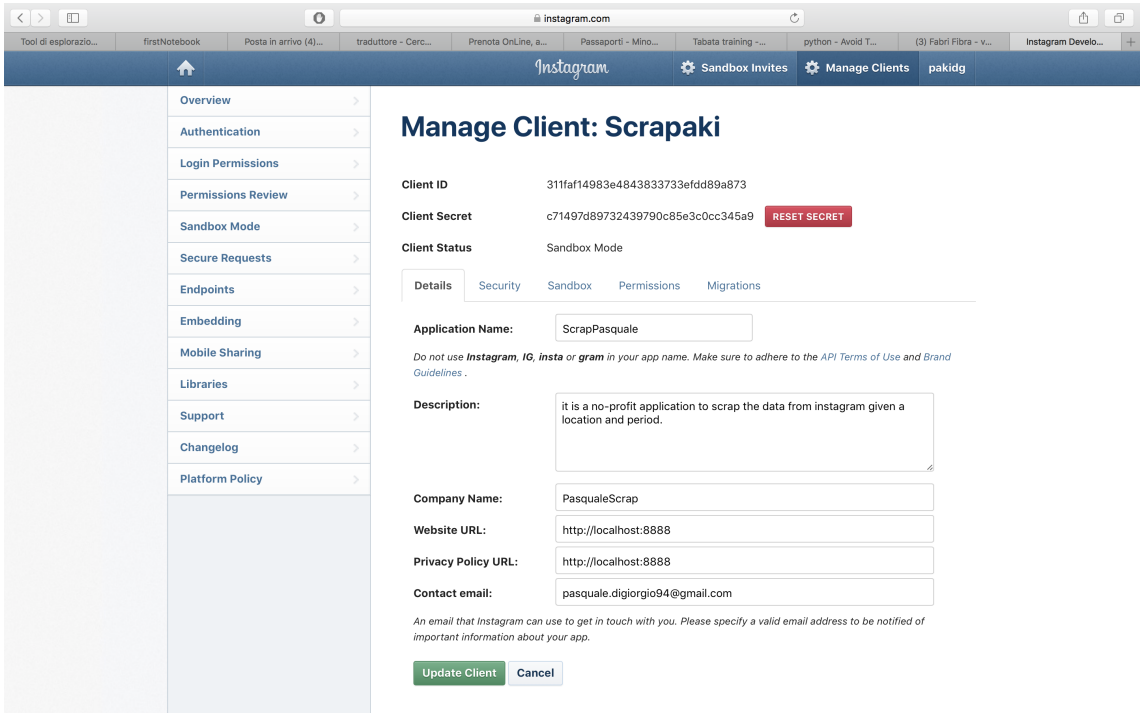

Fig. 3.10. Accessing to Instagram API

The system for the autentificato and the owning of the token is different form the other two social networks. In this case is necessary follow the procedure explained below:

Direct the user to our authorization url using this link

```
https://api.instagram.com/oauth/authorize/?client_id=CLIENT-ID&
    redirect_uri=REDIRECT-URI&response_type=code
```

The output contains URL that you copy and paste into the address bar of a web browser. Following that link, you'll be redirected to log into your Instagram account and finally redirected again to the Website URL you declared when registering your client, but the URL has a special token in the form of ?code=... appended to it.

3.6.1. Making a Instagram API Requests

After obtaining all the necessary we are able to making request like in the example in the code 3:

- *Code 3*

```
import requests # pip install request
```

```
import json # pip install json
from instagram.client import InstagramAPI # pip install InstagramAPI
    -sdk

CLIENT_SECRET = ""

payload = dict(client_id= "",
               client_secret= CLIENT_SECRET,
               grant_type="authorization_code",
               redirect_uri="http://localhost:8888",
               code="")

response = requests.post('https://api.instagram.com/oauth/
    access_token', data=payload)

ACCESS_TOKEN = response.json()['access_token']

Request to obtain my post .


post = requests.get('https://api.instagram.com/v1/users/self/media/
    recent/?access_token='+ACCESS_TOKEN)

recent_post = post.json()

outfile = "InstaJson.json"

out_file = open(outfile,"w")

out_file.write(json.dumps(recent_post, indent=1))

out_file.close()
```

We can see the results in the figure 4.11:

Out[32]:

	attribution	caption	carousel_media	comments	created_time	filter	id	images	likes	
0	None	{u'created_time': u'1539850104', u'text': u'¿C...	NaN	{u'count': 5}	1539850104	Lark	189272231643444230 3_1907483119	{u'low_resolution': {u'url': u'https://sconten...	{u'count': 42}	https://v
1	None	{u'created_time': u'1539157229', u'text': u'Fo...	[{u'images': {u'low_resolution': {u'url': u'ht...	{u'count': 2}	1539157229	Normal	188691006464916793 5_1907483119	{u'low_resolution': {u'url': u'https://sconten...	{u'count': 95}	https://v
2	None	{u'created_time': u'1537773258', u'text': u'I ...	NaN	{u'count': 1}	1537773258	Normal	187530047318844075 6_1907483119	{u'low_resolution': {u'url': u'https://sconten...	{u'count': 96}	https://
3	None	{u'created_time': u'1535907213', u'text': u'Fr...	NaN	{u'count': 3}	1535907213	Nashville	185964694647060161 6_1907483119	{u'low_resolution': {u'url': u'https://sconten...	{u'count': 40}	https://wwv
4	None	None	[{u'images': {u'low_resolution': {u'url': u'ht...	{u'count': 0}	1535633488	Normal	185735078041657702 3_1907483119	{u'low_resolution': {u'url': u'https://sconten...	{u'count': 65}	https://
5	None	{u'created_time': u'1535450301', u'text': u'Un...	[{u'images': {u'low_resolution': {u'url': u'ht...	{u'count': 1}	1535450301	Normal	185581409656661833 9_1907483119	{u'low_resolution': {u'url': u'https://sconten...	{u'count': 60}	https:/
6	None	{u'created_time': u'1534775859', u'text': u'si...	NaN	{u'count': 0}	1534775859	Normal	185015646321823545 6_1907483119	{u'low_resolution': {u'url': u'https://sconten...	{u'count': 66}	https://w
7	None	{u'created_time': u'1534347973', u'text': u'Pa...	NaN	{u'count': 3}	1534347973	Normal	184656709744328702 1_1907483119	{u'low_resolution': {u'url': u'https://sconten...	{u'count': 57}	https://w
8	None	{u'created_time': u'1534004866', u'text': u'É ...	NaN	{u'count': 9}	1534004866	Mayfair	184368890438198229 0_1907483119	{u'low_resolution': {u'url': u'https://sconten...	{u'count': 71}	https://www
9	None	{u'created_time': u'1532254144', u'text': u'Ag...	NaN	{u'count': 9}	1532254144	Normal	182900278514080873 8_1907483119	{u'low_resolution': {u'url': u'https://sconten...	{u'count': 81}	https
10	None	{u'created_time': u'1530284272', u'text': u'[illegible]...	NaN	{u'count': 2}	1530284272	Normal	181247829849697907 9_1907483119	{u'low_resolution': {u'url': u'https://sconten...	{u'count': 55}	https://w
11	None	{u'created_time': u'1525533341', u'text': u'Sc...	NaN	{u'count': 2}	1525533341	Slumber	177262460551804051 2_1907483119	{u'low_resolution': {u'url': u'https://sconten...	{u'count': 75}	https://w
12	None	{u'created_time': u'1524475843', u'text': u'Go...	[{u'images': {u'low_resolution': {u'url': u'ht...	{u'count': 0}	1524475843	Normal	176375366570314512 8_1907483119	{u'low_resolution': {u'url': u'https://sconten...	{u'count': 67}	https://
13	None	{u'created_time': u'1524329573',	NaN	{u'count': [illegible]	1524329573	Normal	176252666995829549 1_1907483119	{u'low_resolution': {u'url':	{u'count': [illegible]	https://wv

Fig. 3.11. Results of simple Instagram API calls

3.6.2. Limits

In the various examples shown we were able to make requests that allowed us to obtain only data concerning the account of the author, because instagram has strict privacy restrictions regarding the use of its API. In order to have fully access at Instagram content, you will need to submit your application for review and approval. Once reviewed, you will only be able to request users the Permission Scopes for which your app was approved. Because of this, your application may not be able to use some API endpoints unless the corresponding permissions were reviewed and approved.

3.7. Comparison of the three

In this research, we analyzed all three platforms from the point of view of data scraping. Finally, we focused on what made us get more data for what we needed nowadays. Since July 31, 2018 facebook has restricted the conditions on privacy, almost completely

depriving us of obtaining data about pages or people with public profiles except those who give permission explicitly, for more information see the privacy policy check here [https://developers.facebook.com/policy].

Instagram instead, becoming part of the Facebook family in recent years, is based on a very similar policy that concerns privacy despite there are many public content. In fact, as you can see just entering the main page of "instagram developer" tells you that from April 2014, the following [26] capabilites have been disabled and that from December 2011, even the so-called "public content" [27] have been deprecated, thus denying us the possibility of accessing public data that we would have needed for this research. According to many developers and also in my opinion all these restrictions are due, in part, to the Cambrige Analytica [28] issue.Taking these considerations into account, we decided to focus only on Twitter as it was the one that provided us with the most data.

4. OBTAIN GEOLOCALIZED DATA FROM SOCIAL NETWORKS

The aim of this chapter is to show how the different solutions proposed in this work have been implemented in order to extract data from the Social Networks analyzed with a precise geolocation thanks to geographical coordinates. Specifically, here we show solutions that take data already present in social media and hence not in streaming. In addiction, we will then show the different limitations imposed by the policies of the various Social Networks.

4.1. Scrap Geo-Tag Post with Facebook

In order to retrieve the geo localized posts from facebook and tweeter we need to perform different tasks. Firstly, we need the access token and when it is obtained, as shown in section 3.4.1, it is possible to use it to make API calls to the facebook API via python. What has been done for this task consisted in creating a script that give geographical coordinates in order to find all the posts in the range of these coordinates. For the *Code 4* a *Facebook SDK for Python* library which contains the class *facebook.GraphAPI* has been used. The latter also contains a client for the Facebook Graph API. The Graph API is made up of the objects or nodes in Facebook (e.g., people, pages, events, photos) and the connections or edges between them (e.g., friends, photo tags, and event RSVPs). This client provides access to those primitive types in a generic way [29] . In the *Code 4* we use the function *request* with the filtering *"search"* which allows us to obtain, on the basis of search criteria, the posts that can be captured within certain coordinates. The results are then saved to a json file or printed on the screen.

- *Code 4*

```
import facebook # pip install facebook-sdk
import json # pip install json
import pandas as pd # pip install Panda

ACCESS_TOKEN = "*not mentioned due privacy reason*"

# A helper function to pretty-print Python objects as JSON
def pp(o):
    print(json.dumps(o, indent=1))

# Create a connection to the Graph API with your access token
g = facebook.GraphAPI(ACCESS_TOKEN, version='3.2')

```

```
out_file = "facebookLoc.json"

out_file = open(out_file,"w")

jsonInfo3 = g.request("search", {'type': 'place', 'center': '
    40.4165000, -3.7025600', 'fields': 'name, location, people, post
    , friends'})

out_file.write(json.dumps(jsonInfo3, indent=1))

out_file.close()

# Get the connections to an ID
# Search for a location, may require approved app
pp(g.request("search", {'type': 'place', 'center': '40.4165000,
    -3.7025600', 'fields': 'name, location, people, post, friends'})
    )

data = json.load(open('facebookLoc.json'))

# df = pd.DataFrame(data["data"])
```

A sample result obtained from this code is the one presented here below. The data have been managed in order to take information by the places present on facebook in the neighborhood of the entered coordinates, but nothing that concerns public posts of people, for privacy reasons. Starting from July 2018, the code here shown contains the information obtained and this is the maximum that we can get as data with a geolocation with the Facebook API.

```
{
 "data": [],
 "summary": {
  "total\_count": 2392
 }
}
{
 "paging": {
  "cursors": {
   "after": "MjQZD"
  },

 },
 "data": [
  {
   "id": "200409866662395",
   "name": "Madrid, Spain",
   "location": {
```

```
      "latitude": 40.4,
      "city": "Madrid",
      "longitude": -3.68333,
      "country": "Spain"
     }
    },
    {
     "id": "221000601260302",
     "name": "Puerta del Sol Madrid",
     "location": {
      "city": "Madrid",
      "zip": "28013",
      "country": "Spain",
      "longitude": -3.7035047838334,
      "street": "Puerta del Sol",
      "latitude": 40.417115338754
     }
    },
    {
     "id": "408745305981456",
     "name": "Community of Madrid",
     "location": {
      "latitude": 40.5,
      "zip": "M",
      "longitude": -3.6666666666667
     }
    },
    {
     "id": "485413701651902",
     "name": "Sol (Madrid)",
     "location": {
      "city": "Madrid",
      "zip": "16415",
      "country": "Spain",
      "longitude": -3.70388889,
      "street": "Avenida Dr. Manuel Jarabo, 14",
      "latitude": 40.41666667
     }
    },
    {
     "id": "208421126617422",
     "name": "Plaza del Sol",
     "location": {
      "city": "Madrid",
      "zip": "28013",
      "country": "Spain",
      "longitude": -3.7024,
      "street": "Plaza de la Puerta del Sol",
      "latitude": 40.41684
     }
    },
```

```
    {
     "id": "876961292367213",
     "name": "Madrid City",
     "location": {
      "city": "Madrid",
      "zip": "28921",
      "country": "Spain",
      "longitude": -3.6898198015885,
      "street": "Paseo Castilla",
      "latitude": 40.406702241129
     }
    },
    {
     "id": "10150290777338362",
     "name": "El Retiro - Jardines del Buen Retiro de Madrid",
     "location": {
      "city": "Madrid",
      "zip": "28009",
      "country": "Spain",
      "longitude": -3.6843369069342,
      "street": "Plaza de la Independencia, s/n",
      "latitude": 40.416726213893
     }
    },
    {
     "id": "152454045334590",
     "name": "Palacio Real de Madrid, Espa\u00f1a",
     "location": {
      "city": "Madrid",
      "zip": "28013",
      "country": "Spain",
      "longitude": -3.71307,
      "street": "Espa\u00f1a",
      "latitude": 40.41822
     }
    },
    {
     "id": "1992503187660492",
     "name": "Madrid - Espanha",
     "location": {
      "city": "Madrid",
      "zip": "28013",
      "country": "Spain",
      "longitude": -3.7053434375486,
      "street": "Madrid, Espa\u00f1a",
      "latitude": 40.420149600521
     }
    },
    {
     "id": "527575860600246",
     "name": "Gran V\u00eda",
```

```
     "location": {
      "city": "Madrid",
      "zip": "28004",
      "country": "Spain",
      "longitude": -3.7030485769043,
      "street": "centr\u00f3 comercial gran v\u00eda",
      "latitude": 40.420084942779
     }
    },
    {
     "id": "1619319278162442",
     "name": "Apple Puerta del Sol",
     "location": {
      "city": "Madrid",
      "zip": "28013",
      "country": "Spain",
      "longitude": -3.7022209,
      "street": "Plaza de la Puerta del Sol, 1",
      "latitude": 40.4168038
     }
    },
    {
     "id": "1487132454924396",
     "name": "Puerta del Sol",
     "location": {
      "city": "Madrid",
      "zip": "28012",
      "country": "Spain",
      "longitude": -3.7033332547623,
      "street": "Puerta del Sol",
      "latitude": 40.416979822785
     }
    },
    {
     "id": "1148013821914644",
     "name": "Plaza Major, madrid",
     "location": {
      "city": "Madrid",
      "zip": "28005",
      "country": "Spain",
      "longitude": -3.707414044,
      "street": "Traves\u00eda de Bringas",
      "latitude": 40.415477656
     }
    },
    {
     "id": "1793968507599585",
     "name": "Puerta del Sol - Madrid Centro",
     "location": {
      "city": "Madrid",
      "zip": "28014",
```

```
      "country": "Spain",
      "longitude": -3.69964,
      "street": "10  Calle de Echegaray",
      "latitude": 40.41563
     }
    },
```

4.2. Scrap Geo-Tag Post with Instagram

Recalling what we have seen before concerning the Instagram's API, we remember that they allow to obtain data only about your own account, unless you submit the application for the use of "public_contents". In fact, since June 2016, for privacy reasons the media/search endpoint in Sandbox mode was limited to return just the media you uploaded from that location and nothing else. Therefore, in order to have access to the public content published by others, you need to submit your application to Instagram that requires approval for the Live mode [30]. The only thing that we have been able to managed in terms of geographical locations, are the places present within the posts that have been uploaded by the person that is using the API we are talking about. A simple example we can see in the Fig. 5.1 .

likes	link	location	tags	type	user	user_has_liked	users_in_photo
count': 42}	https://www.instagram.com/p/BpETOsxh4g_g0c0pXP...	{u'latitude': 40.41842, u'name': u'Círculo de ...	[]	image	{u'username': u'pakidg', u'profile_picture': u...	False	[]
count': 95}	https://www.instagram.com/p/BovprTeheQ_1Z1vmC6...	None	[]	carousel	{u'username': u'pakidg', u'profile_picture': u...	False	[]
count': 96}	https://www.instagram.com/p/BoGZ9g3hNq0riXde3y...	{u'latitude': 40.4, u'name': u'Madrid, Spain',...	[]	image	{u'username': u'pakidg', u'profile_picture': u...	False	[]
count': 40}	https://www.instagram.com/p/BnOywuEBSOQHPnQhVD...	None	[]	image	{u'username': u'pakidg', u'profile_picture': u...	False	[]
count': 65}	https://www.instagram.com/p/BnGorGyBk3_74j8Ts0...	{u'latitude': 39.8, u'name': u'Leuca', u'longi...	[]	carousel	{u'username': u'pakidg', u'profile_picture': u...	False	[]
count': 60}	https://www.instagram.com/p/BnBLRabBAjjtr021yK...	None	[]	carousel	{u'username': u'pakidg', u'profile_picture': u...	False	[]
count': 66}	https://www.instagram.com/p/BmtE4AxBBhAd1hzMRP...	None	[]	image	{u'username': u'pakidg', u'profile_picture': u...	False	[]
count': 57}	https://www.instagram.com/p/BmgUv3ZB__tuyI5HWU...	{u'latitude': 41.70463, u'name': u'Monte Sant'...	[]	image	{u'username': u'pakidg', u'profile_picture': u...	False	[]
count': 71}	https://www.instagram.com/p/BmWGUpOBGZSHzzkdbX...	{u'latitude': 40.5379308811, u'name': u'Puglia...	[bentornatacaviglia, buddha, zenphylosophy, day1]	image	{u'username': u'pakidg', u'profile_picture': u...	False	[]
count':	https://www.instagram.com/p/Blh7Fe1hJwiol4uffa...	{u'latitude': 43.6681077875,	[]	image	{u'username': u'pakidg',	False	[]

Fig. 4.1. Simple result obtained on information about the places of the posts of Instagram

For these reasons we have not been able to develop any solution to extrapolate geolocalized data with Instagram and we decided to not proceed further with this social media because it doesn't match the objective of this thesis.

4.3. Scrap Geo-Tag Post with Twitter

The last social media under investigation is tweeter and in with the aim of understanding how twitter APIs work it is necessary to look at the main elements of this platform.The *Tweets*, that are the fundamental entity of this social, are associated with two important metadata:

1. TWEET ENTITY -> User mentions, hashtag, url and media;
2. PLACES -> Locations in the real world.

In this research project one of the main objective focuses on the second metadata. With this in mind it has been created a script to get the data of the geolocalized tweets, as shown in the Code 5. The different steps required to develop this code made necessary to use a specific twitter library created for python [31]. This library provides a pure Python interface for the Twitter API. Firstly, we create an object that contains our four key, secondly we use it to make a API call to search for all post in a specific Area. Finally, we save the result in a .csv file and thus it is possible to view in a better way what kind of data it pulled out. Code 5 was created with the help of a set of code's examples that we can find here [32].

- *Code 5*

```
import twitter # pip install twitter-sdk
import csv # pip install csv
import json # pip install json

latitude = raw_input("insert latitude: \n")
longitude = raw_input("insert longitude: \n")
max_range = raw_input("insert search range in kilometres: \n")
num_results = 200

latitude = float(latitude)
longitude = float(longitude)
max_range = float(max_range)

outfile = "LondonCsv.csv"

#I don't put my real keys due to privacy reason

CONSUMER_KEY = ''
CONSUMER_SECRET = ''
OAUTH_TOKEN = ''
OAUTH_TOKEN_SECRET = ''
auth = twitter.oauth.OAuth(OAUTH_TOKEN, OAUTH_TOKEN_SECRET,
```

```
                                CONSUMER_KEY, CONSUMER_SECRET)
twitter_api = twitter.Twitter(auth=auth)

# open a file to write (mode "w"), and create a CSV writer object
csvfile = open(outfile, "w")
csvwriter = csv.writer(csvfile)

# add subject to our CSV file

row = [ "user", "text", "latitude", "longitude" ]
csvwriter.writerow(row)

result_count = 0

while result_count <  num_results:

    query = twitter_api.search.tweets(q = "", geocode = "%f,%f,%dkm"
        % (latitude, longitude, max_range))

    for result in query["statuses"]:

        if result["geo"]:
            result_count +=1

            user = result["user"]["screen_name"]
            text = result["text"]
            text = text.encode('ascii', 'replace')
            latitude = result["geo"]["coordinates"][0]
            longitude = result["geo"]["coordinates"][1]

            # now write this row to our CSV file
            row = [ user, text, latitude, longitude ]
            csvwriter.writerow(row)
            result_count += 1
            last_id = result["id"]


csvfile.close()
```

A simple result obtained from this code is shown in Fig. 5.2. What we can immediately notice is that using these APIs, you can get the data within a certain range of coordinates, but most of the time the data obtained contains the same informations already get.

	user	text	latitude	longitude
0	MakeOverEssex	?? Good Morning Monday! It?s gonna be an amazi...	51.507115	-0.127318
1	BowmanCourtney_	?????????? \nSo proud of what we have created ...	51.510947	-0.134224
2	NellyLeyShop	Old engraving work done by myself when I was ...	51.507115	-0.127318
3	NellyLeyShop	Old engraving work done by myself when I was ...	51.507115	-0.127318
4	NellyLeyShop	Old engraving work done by myself when I was ...	51.507115	-0.127318
5	NellyLeyShop	Old engraving work done by myself when I was ...	51.507115	-0.127318
6	NellyLeyShop	Old engraving work done by myself when I was ...	51.507115	-0.127318
7	NellyLeyShop	Old engraving work done by myself when I was ...	51.507115	-0.127318
8	NellyLeyShop	Old engraving work done by myself when I was ...	51.507115	-0.127318
9	NellyLeyShop	Old engraving work done by myself when I was ...	51.507115	-0.127318
10	NellyLeyShop	Old engraving work done by myself when I was ...	51.507115	-0.127318
11	NellyLeyShop	Old engraving work done by myself when I was ...	51.507115	-0.127318
12	NellyLeyShop	Old engraving work done by myself when I was ...	51.507115	-0.127318
13	NellyLeyShop	Old engraving work done by myself when I was ...	51.507115	-0.127318
14	NellyLeyShop	Old engraving work done by myself when I was ...	51.507115	-0.127318
15	NellyLeyShop	Old engraving work done by myself when I was ...	51.507115	-0.127318
16	NellyLeyShop	Old engraving work done by myself when I was ...	51.507115	-0.127318
17	NellyLeyShop	Old engraving work done by myself when I was ...	51.507115	-0.127318
18	NellyLeyShop	Old engraving work done by myself when I was ...	51.507115	-0.127318
19	NellyLeyShop	Old engraving work done by myself when I was ...	51.507115	-0.127318
20	NellyLeyShop	Old engraving work done by myself when I was ...	51.507115	-0.127318
21	NellyLeyShop	Old engraving work done by myself when I was ...	51.507115	-0.127318
22	NellyLeyShop	Old engraving work done by myself when I was ...	51.507115	-0.127318
23	NellyLeyShop	Old engraving work done by myself when I was ...	51.507115	-0.127318
24	NellyLeyShop	Old engraving work done by myself when I was ...	51.507115	-0.127318

Fig. 4.2. Simple result obtained on information about the places of the tweets in London

4.3.1. Limits

The previous code has a limitation in terms of numbers of API calls that we are able to do consistently. Indeed, if we try to obtain data using the same function, we get the following error after 100 calls:

```
{"errors":[{"message":"Rate limit exceeded","code":88}
```

This is due to the fact that Twitter sets a limit on how many times it can be used in an

hour. This limit, applied to your Twitter account rather than the applications which make the calls to the API, results in such an error. For instance if you have 100 API calls per hour in total regardless of which Twitter applications you are using, they are NOT 100 API calls per application. It is also important to note that this limit only applies to 3rd parts Twitter applications, in fact the twitter.com website does not use it's own API (doesn't seem particularly fair) and therefore has no limits. So what does constitutes an API call? Strictly speaking, every operation which communicates with Twitter is an API call. What we really need to know is which API calls have an impact on the 100 calls limit. The simplest way to think about this is focusing on the fact that every call to the Twitter API that requests data will count towards your limit. Sending data to Twitter (posting), such as posting an update or a direct message, favoriting a tweet, unfollowing or following a user, does not count towards the limit and you can continue to do so even when your rate limit has been exceeded.

4.4. Conclusion: how to proceed?

As we can see, there are many limitations imposed by the APIs of these social to obtain data already stored with a precise geolocation. The only APIs that have provided us with concrete data are those of Twitter, but even in this case there is a great limitation: to obtain data that refer to an earlier period of seven days from today, you must open a premium account and pay a monthly subscription [33] . For these reasons, since seven days are too few to make real predictions, it has been decided to continue with this project using unofficial Twitter APIs that allow you to get streaming data without any kind of limitation.

5. METHODOLOGY

In this chapter we will discuss how we dealt with the main objective of this work, which is the methodology applied and with method have been chosen and why. Firstly we will talk about how identify anomalies that may occur during a given period in a specific city, afterwards, we will point out how the behavior of the crowd changes in the different cities analyzed. The scenario is the follows: we analyze the location of people during a predetermined period in a specific city based on their coordinates. As a matter of fact, this position changes during the same day and also day by day, but even if this changes, usually customs of people are almost constant and hence it remains the same if we analyze a certain period. This kind of approach can help us identify anomalies.

We split the problem in to main groups and we here identify the main steps of the methodology.

- Develop an algorithm that allows to obtain data in real time about tweets posted by people in a real and precise location identified by geographical coordinates.
- Discretize the values obtained in cells in which each cell represented a part of the city, as a grid.
- Monitor the trend of the crowd in the selected coordinates measuring an Entropy properly defined.
- Check if there are any anomalies and compare the result obtained with that obtained in the same period in other cities.

The cities considered for this experiment are classified in descending order according to the use of twitter and the two test cases are Madrid and Rome. The Fig. 6.1 shows the steps followed during the development of this individual research project. The staring point, in the red circle, cannot be different from a literature review analysis. It is necessary to contextualize the problem in order to deal with it. If any other work exists on the same subject it is worthy study it and discuss it in a critical way. Found out the strength, evaluate the results, take what is considered positive and delate what is considered useless of not properly appropriate. Therefore, the Python tool able to scraping data from twitter required care and attention because of its importance. Nothing but a line by line approach could have been more appropriate. Particular attention must also be given to the part where a code is developed to see the distribution of tweets on the map. Finally, the goal is to define a methodology that suits the problem and is specifically designed for it.

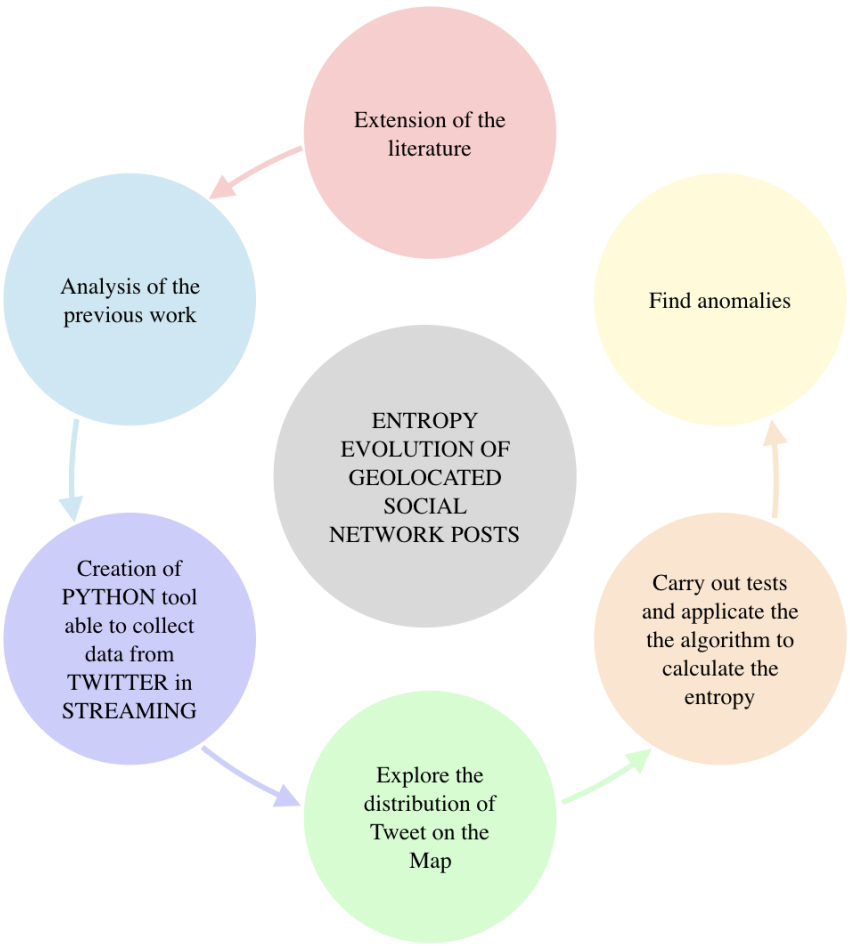

Fig. 5.1. Milestones of the current methodology that represent the steps followed during this project. Made by the author.

5.1. Analysis of Previous Work

The first step is the analysis and discussion of the previous works. Among them, one of the most remarkable work is the research conducted by the Department of Telematic Engineering of the University Carlos III of Madrid in collaboration with the School of Telecommunications Engineering of the University of Vigo [34]. This latter is focused on the LSBN data collecting from instagram to evaluate the entropy related to different cities, underlying any detected anomaly. The aim of this thesis project is to improve the mentioned previous work, firstly by developing an enhanced algorithm with different features for the calculation of entropy and then test it with a wide range of applications, cities and temporal intervals.

5.2. Scraping Geo-Tag tweets Streaming

In this section we analyze the code that has been written in order to obtain data from twitter in Streaming. Twitter provides REST APIs [35] and we can use them to interact with their services. There is also a Python-based clients [36] does the same function and we can use. In particular, the Tweepy [37] library has been used to develop this project and we also followed the guide [38] of Marco Bonzanini. As a first step we had to create an app that interacts with the twitter API through the process explained in chapter three, in particular, we need to point our browser to http://apps.twitter.com, log-in to Twitter and register a new application. We will receive a consumer key and a consumer secret: these are application settings that should always be kept private. From the configuration page of our app, we can also require an access token and an access token secret. Similarly to the consumer keys, these strings must also be kept private: they provide the application access to Twitter on behalf of your account, since both of them must be kept private we save them in a file from which they will be taken each time by the algorithm used, as we can see in *Code 6*.

- *Code 6*

```
with open('twitter_credentials.json') as cred_data:
    info = json.load(cred_data)
    consumer_key = info['CONSUMER_KEY']
    consumer_secret = info['CONSUMER_SECRET']
    access_token = info['ACCESS_KEY']
    access_secret = info['ACCESS_SECRET']
```

Once we get the four keys, in order to authorize our app to access Twitter on our behalf, we need to use the OAuth interface, indeed Tweepy supports OAuth authentication discussed in section 3.3.4. Authentication is handled by the tweepy.AuthHandler class. The next step, performed by the code in *Code 7* is the creation of an instance of OAuthHandler to which we will pass our *consumer_key* and *consumer_secret* obtained in the previous step. Now, in order to receive the authorization from twitter to using its functions, we have also to set *access_token* and *access_secret*. Therefore, now that we have our OAuthHandler equipped with an access token, we are ready for scraping data from twitter. The next operation is necessary to extended the *StreamListener()* for customize how the incoming data are processed, (see *Code 8*) and get all the tweets streaming corresponding to a specific geographical area. To do this task we use the filter function *location* of the API, which given a latitude and a longitude in returns of all the tweets that are posted within that geographical area start from the moment in which the script is launched. We inserted this part of code in an infinite loop, to allow the code to run without interruption and to interrupt only in case of "keyboard_interrupt", i.e. stoped by us.

- *Code 7*

```
while True:
    try:
        auth = OAuthHandler(consumer_key, consumer_secret)
        auth.set_access_token(access_token, access_secret)
        twitterStream = Stream(auth, listener())
        twitterStream.filter(locations
            =[-3.89889846,40.2744000215,-3.504458569,40.5539033977])
    except KeyboardInterrupt:
        file.seek(-1, os.SEEK_END)
        file.truncate()
        file.write("]")
        file.close()
        sys.exit()
        print count + "downloaded tweets"
    except:
        continue
```

In the *Code 8* we show how we customized the way to process the incoming data. Specifically, we have created a class *listener(StreamListener)*, in which every time we get a new tweet we transform it from json format to python object, afterwards we filter the data in the following way:

- We check if the tweet contains the attribute "geo", which means that the tweet contains the real position and not the one inserted by the user without actually being there at that moment. More information about the difference between these two positions can be found here [39];
- If the first condition is verified then let's proceed with saving the tweet inside a file as an array.

Inside the class is also defined a function that, in case of any error during this process, will print us what kind of error it is.

- *Code 8*

```
file = open("StreamingRome.json", "w")
count = 0
file.write(   [")
file = open("StreamingMadrid.json", "w")

count = 0
file.write("[")
class listener(StreamListener):

    def on_data(self, data):
```

```
11          global count
12
13          #if count <= val:
14          json_data = json.loads(data)
15          coords = json_data["geo"]  # collect only tweets with
               geolocalization attribute
16          if coords is not None:
17              file.write(json.dumps(json_data,indent=4))
18              count += 1
19              file.write(",")
20
21          def on_error(self, status):
22          print status
```

5.3. Entropy Calculation

In this session we will explain step by step what has been done to calculate entropy and we intend as entropy in the case studied within this work. Firstly, it will be shown how the geographical area has been discretized, then it will be explained what it is and how the algorithm works. Finally, the code we wrote to apply the method to the data obtained with the code explained in the previous session will be commented line by line.

5.3.1. Discretize of the values

The final purpose of the entropy calculation of this project is to analyze the movement of the crowd during a given period. In order to do this we need a set of values that indicate the crowd in different areas of the city. In this project the area of Madrid and Rome have been taken into consideration, in Fig. 6.2 we can see how it has been selected and how large is the area taken into account.

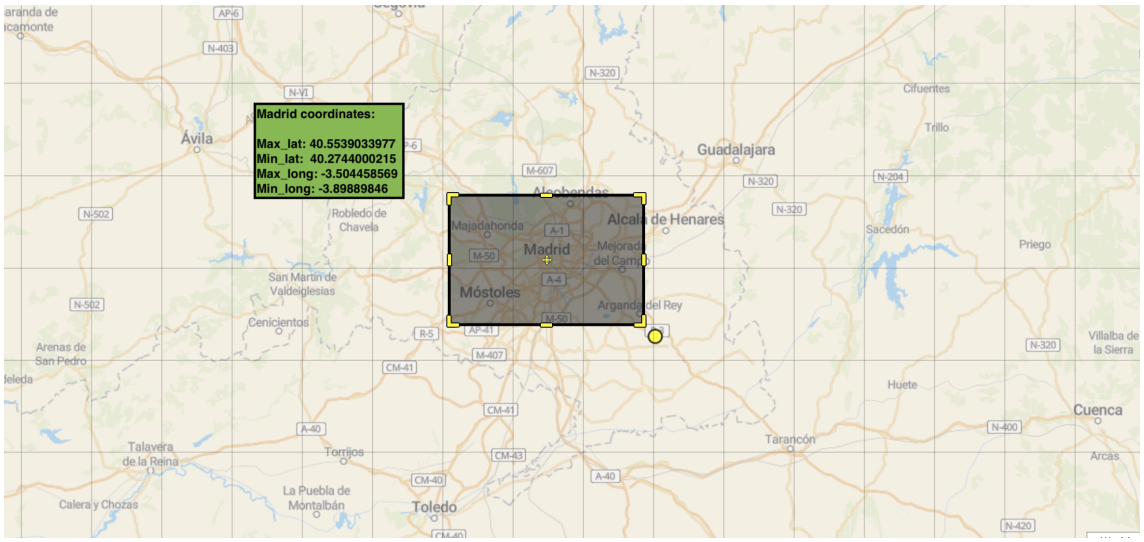

Fig. 5.2. Madrid Coordinates range

with the aim of implementing the map into the code it is necessary to generate a grid of points that will be used to locate the tweets. We have therefore take the coordinates of the city center into the symbolic domain, which allows us to computing the entropy with the sequence explained later on. The city is divided into non-overlapping NxN cells of the same size, as we can see in Fig. 6.3. Each cell is then labeled with a symbol. The position of the center at each time interval is identified by the symbol of the cell that encloses that position. In this way we are able to map all the area and precisely locate tweets and calculate the entropy associated with them.

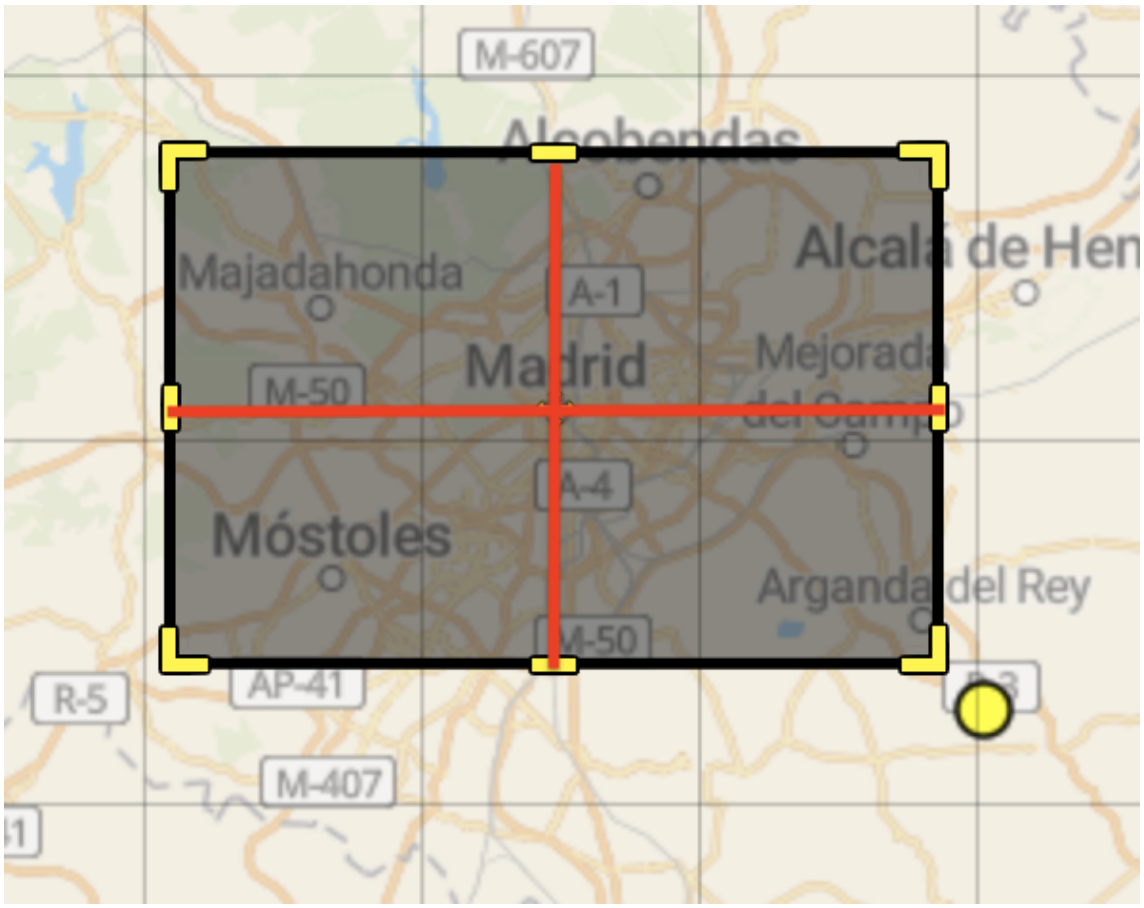

Fig. 5.3. Madrid Coordinates range with grid

The calculation of $n-th$ cell is defined as it follows. We can see that it is a squared grid with the origins in the point (x_0, y_0) that increase in the x and y direction with a step equal to total length of the area selected divided by the number of points desidered. Obviously, the more is the number of point, the more accurate is the discretization and the consequent location of tweets and entropy. The $i-th$ point of the grid is obtained with the following system:

$$\begin{cases} x_i = x_0 + \dfrac{L_x}{N} i \\ y_j = y_0 + \dfrac{L_y}{N} j \end{cases} \tag{5.1}$$

where L_x and L_y are the total length in the two direction of the grid and N is the number of element chosen for the grid.

5.3.2. Entropy Calculation: Shannon equation

In order to implement our solution the method shown in the [34] has been used. We calculate the entropy variation with this sequence and then it is necessary to evaluate

how the crowd moves during the chosen period. However, large deviations from the expected uncertainty value can potentially reveal unexpected events. One way to measure the expected uncertainty of a sequence of symbols belonging to an L-alphabet is through Shannon's concept of information theory of entropy. We will now introduce the concept of entropy and its practical interpretation. A wider review on this topic can be found in [40].

Let X be a discrete random variable taking values on an alphabet L, $|L|$ being the cardinality of the alphabet, with Probability Mass Function (PMF) $\Pr(X = l) = p(l), \forall l \in L$. Then, the Shannon entropy of X can be written as:

$$H = -\sum_{l \in L} p(l) \log_2 p(l) \tag{5.2}$$

where the base two logarithm denotes that the resulting entropy value is measured in bits and where $p(l)$ is the probability of symbol l. Paying attention to the practical meaning of entropy, H measures the expected "surprise" or uncertainty enclosed by the random variable X. Since the probability mass function $p(l)$ is not available, and our data were not an infinite sequence of symbols, we approximate it by a maximum likelihood estimator based on the observable data:

$$p(l, i) = \frac{N_{l,i}}{i}, 0 \leq i \leq n \tag{5.3}$$

where $N_{l,i}$ is the number of appearance of location l in the sequence from the beginning up to time interval i and n is the total number of time intervals. Applying this to the entropy formula, for each time interval, i, we have:

$$H(i) = -\sum_{l \in L} p(l, i) \log_2 p(l, i) \tag{5.4}$$

As we will explain later, at each interval, we will calculate the entropy from the beginning of the sequence up to that interval, H, and also the entropy considering just the last *win* symbols of the sequence, H_{win}, with *win* ranging from 2 weeks–2 months. For H_{win} we will use Equation 6.4, but with $p(l, i)$ being:

$$p(l, i) = \frac{N_{wl,i}}{win}, win \leq i \leq n \tag{5.5}$$

where $N_{wl,i}$ is the number of appearances of location l in the last *win* symbols of the sequence (from $i - win + 1$–i). Finally, we inspect the values of the entropy calculated at each time interval i, $H(i)$, or $H_{win}(i)$, depending on whether we consider the entropy from the beginning of the last win symbols, and label as potential anomalies those samples with higher entropy differences with respect to the previous value.

5.3.3. Experiment and Parameter Selection

In this session, the code used to calculate the entropy of the data obtained is explained line by line. As a first step, as it possible to see in *Code 9*, we take the data we have collected and transform it using the Panda library [41], which allows very easily to explore files of this type by converting strings into lists and dictionaries.

- *Code 9*

```
import matplotlib
import math
import scipy.stats
import json
import pandas as pd
import matplotlib.pyplot as plt
import math
import uc3mEntropy
from pandas.io.json import json_normalize
import numpy as np
from datetime import datetime
import matplotlib.patches as mpatches


with open('StreamingRoma.json') as dati:
    contents = dati.read()
    new = json.loads(contents)

#para trasformar en Panda series
nycphil = json_normalize(new)
```

Secondly, (see*Code 10*) the formula explained in the previous section is implemented in order to divide the geographical area in *NxN* cells of the same size. Specifically, we take the coordinates that correspond to the extremes of the area under consideration and we save them as variables, then we scroll the coordinates of all tweets with a loop and for each coordinate (longitude and latidudine) we apply the formula 6.1. Then the values corresponding to the time the post was posted are saved in an array.

- *Code 10*

```
max_lat = 41.9919693871
min_lat = 41.7932623901
max_lng = 12.6650081465
min_lng = 12.3045870488

#para discretizar en n*n cells
```

```
def discretize(row, n):
    return ( int( n * (row[0] - min_lat) / (max_lat - min_lat) ) ) +
         \
           ( n * int( n * (row[1] - min_lng) / (max_lng - min_lng) )
               )

val = []
for i in nycphil["geo.coordinates"]:
    val.append(discretize(i,4))

time = []
for i in nycphil["created_at"]:
    time.append(i)
```

Finally, as it is possible to see in the *Code 11*, the shannon algorithm is applied for each value obtained from the discretization in cells. Similarly, in the *Code 12* is calculated the entropy with a window that moves from time to time. In this specific case, it is calculated the entropy for $1/3$ of the values. The window is calculated as the $n - th$ entropy to which the value in head is eliminated and added the new value in tail, this allows us to have more concrete values.

- *Code 11*

```
#sin ventana
entr = []
entropia = []

count = 0
for j in val:
    #count +=1
    entr.append(j)
    serie = pd.Series(entr)
    entropia.append(uc3mEntropy.scipy_entropy(serie))


df = pd.DataFrame({
    'value': entropia
}, index = time)

line = df.plot.line()
```

- *Code 12*

```
#con ventana
entrVent = []
```

```
entropiaVent = []

i = 0
for j in val:
    i = i + 1
    #voy a eliminar el primero valor asi la ventana se mueve siempre
        mas
    if i > abs(len(val)/3):
        entrVent.pop(0)
        entrVent.append(j)
        serie = pd.Series(entrVent)
        entropiaVent.append(uc3mEntropy.scipy_entropy(serie))
    else:
        entrVent.append(j)
        serie = pd.Series(entrVent)
        entropiaVent.append(uc3mEntropy.scipy_entropy(serie))

df = pd.DataFrame({
    'value': entropiaVent
}, index = time)

line = df.plot.line()
```

5.3.4. Limitations and Advantages

The limitations of using twitter and the API that allowed us to conduct this investigation is that they are very sensitive to policy changes, as we had already seen in the use of other APIs of other social media. We have found that many changes occurred in the last years but fortunately the Tweepy library does not impose many limitations. In fact, it was possible to start several scripts with different search parameters at the same time and let them work in parallel. The only limitation in this sense was the script that collected real-time data from New York, which stopped after only 13 days, causing a "403" error that, according to the official twitter documentation, is caused when the request to contact the end-point was refused or the access not allowed. This code is used when the requests are rejected due to update limits. Most likely, we were in the case "Corresponds with HTTP 403 The user account has been suspended and information cannot be retrieved", as we can see from the official documentation [42]. This is because too many calls to the API are made at the same time with the same beliefs. As a matter of fact, this has been caused by the fact that the data being captured for New York was 10 times larger than the other two.

Another limitation of our approach is the way in which the twitter API provides information on geolocated tweets. The tweet position was obtained from the latitude and longitude where the tweet was posted. However according to a study conducted in [43], during the one-week sampling period of the study, about 20% of the tweets collected showed the user's location with an accuracy of the street level or higher. Many Twitter

users disclosed their physical locations directly through active monitoring of the location or Global Positioning System (GPS) coordinates. However, another 2.2% of all tweets - about 4.4 million tweets a day - has provided "environmental" location data, in which the user may not be aware that he is disclosing his location. Therefore, what emerges is that only a very small part of the total tweets have information about geolocation, and many of these do not provide accurate information. In addition, another work [44] investigated the geographical distribution of tweets and the amount of tweets with precise geolocation in relation to the total. Out of a total of 3977238 of our sample, only 11513 have an exact position instead of the other 27213 that are "tweets with place" [39]. In conclusion, with respect to the 100% of tweets only 0.29% have a precise geolocation, and these are the ones we are interested in. In this work, we are interested in revealing unusual events such as special days or even emergencies, in fact it was noted that during holidays, such as Christmas and New Year, we have an increase in tweets and also a movement of these in different areas of the city than they usually had. In any case, these aspects are still open to further investigation.

6. TESTS AND RESULTS

The purpose of this chapter is to describe the results obtained thanks to the methodology proposed in the previous chapter, their goodness and drawbacks with the aim of commenting and analyzing the features and the issues observed. Due to the limitations and issues discussed in the previous chapter, in the early stages this work has been quite tricky, and some changes have been applied in order to proceed further. Results will be presented and a general statistics analysis of the case will be carried out thanks to the data obtained and the phenomena observed. We will discrete how the distribution of the tweets has been developed, which are the scripts applied and why and finally the entropy analysis will be widely commented.

6.1. Data obtained

The data obtained are saved in a *.json* file type which is a format easy to be handled by python and that allows to work with a huge amount of data. Moreover, python, thanks to the "json" library, offers great support for the manipulation and exploration of these types of files.

In the first stages, the algorithm mentioned in the previous section was launched not only for Madrid and Rome, but also for New York. However, on December 23rd the algorithm break down in the last city causing an error due to too many calls to the API made at the same time with the same beliefs. This is because the data being captured for New York was 10 times larger than the other two and both servers and routine cannot handled them. For this reason, the statistical analysis here proposed has been done only for Madrid and Rome, where we have all the necessary data.

Here is an example of a tweet of Madrid as it has been captured by the code 6,7,8 mentioned in the previously chapter:


```
{
    "quote_count": 0,
    "contributors": null,
    "truncated": true,
    "text": "Repitir\u00eda una y mil veces este viaje. A pesar de
       que ya haya pasado m\u00e1s de un mes, lo recuerdo como si
       hubiera ater\u2026 https://t.co/tvJ4k58TFJ",
    "is_quote_status": false,
    "in_reply_to_status_id": null,
    "reply_count": 0,
    "id": 1072132468625428484,
    "favorite_count": 0,
```

```
    "entities": {
        "user_mentions": [],
        "symbols": [],
        "hashtags": [],
        "urls": [
            {
                "url": "https://t.co/tvJ4k58TFJ",
                "indices": [
                    117,
                    140
                ],
                "expanded_url": "https://twitter.com/i/web/status
                   /1072132468625428484",
                "display_url": "twitter.com/i/web/status/1\u2026"
            }
        ]
    },
    "retweeted": false,
    "coordinates": {
        "type": "Point",
        "coordinates": [
            -3.7081756,
            40.421538
        ]
    },
    "timestamp_ms": "1544451274435",
    "source": "<a href=\"http://instagram.com\" rel=\"nofollow\">
       Instagram</a>",
    "in_reply_to_screen_name": null,
    "id_str": "1072132468625428484",
    "retweet_count": 0,
    "in_reply_to_user_id": null,
    "favorited": false,
    "user": {
        "follow_request_sent": null,
        "profile_use_background_image": true,
        "default_profile_image": false,
        "id": 1151992320,
        "default_profile": false,
        "verified": false,
        "profile_image_url_https": "https://pbs.twimg.com/
           profile_images/1044706544355627014/qtCWojQk_normal.jpg",
        "profile_sidebar_fill_color": "F6F6F6",
        "profile_text_color": "333333",
        "followers_count": 346,
        "profile_sidebar_border_color": "EEEEEE",
        "id_str": "1151992320",
        "profile_background_color": "ACDED6",
        "listed_count": 7,
        "profile_background_image_url_https": "https://abs.twimg.com
           /images/themes/theme18/bg.gif",
```

```
            "utc_offset": null,
            "statuses_count": 19140,
            "description": "92:48. Dicen que se me da bien escribir.\
               nIcf1, rmcf. 25.22.14.24.\nAmante de la fotograf\u00eda.
                \ud83d\udcf7\nComunicaci\u00f3n Audiovisual en la Uex \
               ud83c\udfa5",
            "friends_count": 455,
            "location": "Zafra, Badajoz.",
            "profile_link_color": "038543",
            "profile_image_url": "http://pbs.twimg.com/profile_images
               /1044706544355627014/qtCWojQk_normal.jpg",
            "following": null,
            "geo_enabled": true,
            "profile_banner_url": "https://pbs.twimg.com/profile_banners
               /1151992320/1534455969",
            "profile_background_image_url": "http://abs.twimg.com/images
               /themes/theme18/bg.gif",
            "name": "Sensual Unicorn \u2661",
            "lang": "es",
            "profile_background_tile": false,
            "favourites_count": 13460,
            "screen_name": "laurats99",
            "notifications": null,
            "url": "http://sensualunicorn.blogspot.com",
            "created_at": "Tue Feb 05 20:41:59 +0000 2013",
            "contributors_enabled": false,
            "time_zone": null,
            "protected": false,
            "translator_type": "none",
            "is_translator": false
        },
        "geo": {
            "type": "Point",
            "coordinates": [
                40.421538,
                -3.7081756
            ]
        },
        "in_reply_to_user_id_str": null,
        "possibly_sensitive": false,
        "lang": "es",
        "extended_tweet": {
            "display_text_range": [
                0,
                206
            ],
            "entities": {
                "user_mentions": [],
                "symbols": [],
                "hashtags": [],
                "urls": [
```

```
                {
                    "url": "https://t.co/YXjsYhkhWK",
                    "indices": [
                        183,
                        206
                    ],
                    "expanded_url": "https://www.instagram.com/p/
                       BrNbIlDFTRW/?utm_source=ig_twitter_share&
                       igshid=7njy12buwk11",
                    "display_url": "instagram.com/p/BrNbIlDFTRW/\
                       u2026"
                }
            ]
        },
        "full_text": "Repitir\u00eda una y mil veces este viaje. A
           pesar de que ya haya pasado m\u00e1s de un mes, lo
           recuerdo como si hubiera aterrizado anoche.\n\nHablando
           de viajes, pronto pongo rumbo a una nueva\u2026 https://
           t.co/YXjsYhkhWK"
    },
    "created_at": "Mon Dec 10 14:14:34 +0000 2018",
    "filter_level": "low",
    "in_reply_to_status_id_str": null,
    "place": {
        "full_name": "Madrid, Espa\u00f1a",
        "url": "https://api.twitter.com/1.1/geo/id/206c436ce43a43a3.
           json",
        "country": "Espa\u00f1a",
        "place_type": "city",
        "bounding_box": {
            "type": "Polygon",
            "coordinates": [
                [
                    [
                        -3.889005,
                        40.312071
                    ],
                    [
                        -3.889005,
                        40.643518
                    ],
                    [
                        -3.51801,
                        40.643518
                    ],
                    [
                        -3.51801,
                        40.312071
                    ]
                ]
            ]
```

```
146             },
147             "country_code": "ES",
148             "attributes": {},
149             "id": "206c436ce43a43a3",
150             "name": "Madrid"
151         }
```

As we can see, the metadata collected are of different nature. In order to help us with the analysis of the such a data we used the library called "Pandas" [41], that allows a very easy exploration of these kind of files by converting strings into lists and dictionaries. According to what we have seen in the chapter 3 the metadata behind a single tweet are really a lot compared to what a person can sees. We obtained also the information about the creation data of the post and its geographical coordinates which are represented as a follow:

```
1
2       "coordinates": {
3           "type": "Point",
4           "coordinates": [
5               -3.7081756,
6               40.421538
7           ]
8       }
```

A very interesting thing was we had been able to extract the exact position where the post was posting.

There are quite a few questions we could answer using this dataset:

- Which day of the week has the most tweets?
- At what time of day are people more active on twitter?
- Are the tweets distributed fairly evenly in geographical terms?

6.1.1. Exploring the data

The two *.json* files obtained are of a size of about 300MB. Respectively, 21463 tweets were captured from Rome, while 54460 tweets were captured from Madrid. They were captured during a period from 10/12/2018 to 10/1/2019. Starting from this first data it is possible to understand where the platform is used more in agreement with researches performed in the past. In order to extract the data from the file and answer the questions previously asked, we have read them and afterwards we transformed them into python objects through the .json library. Finally, it is necessary to switch them into Pandas series

in a way that is capable to treat them not as simple lists of strings but as Data Frames, and this task can be done with the follwing codes.

```
with open('StreamingMadrid.json') as dati:
    contents = dati.read()
    new = json.loads(contents)

nycphil = json_normalize(new)
```

The data in Pandas series allow a great amount of analysis. For instance, it is possible to know which days of week results in the most amount of tweet as it follows:

```
nycphil['created_at'] = pd.to_datetime(nycphil['created_at'])

nycphil["created_at"].dt.weekday.value_counts()
```

Day of the week	**Madrid**	**Rome**
Monday	8178	3129
Tuesday	8577	3516
Wednesday	8701	3238
Thursday	7996	2996
Friday	7177	2929
Saturday	6954	2726
Sunday	6877	2929

TABLE 6.1. NUMBER OF TWEETS PER DAYS OF THE WEEK.

The Table 7.1 shows the amount of tweets for each day of the week during the period from 10/12/2018 to 10/1/2019 for both cities analyzed. It is possible to see that both Madrid and Rome's have a similar trend, that vary only by the number of tweets. Indeed, as we can see from the table, in Madrid we have almost three times the number of tweets of Rome with the same trend. Moreover, as for the distribution we have a greater number of tweets between Tuesday and Wednesday, this is because in accordance with the period in which we performed the capture of the tweets, on Tuesday' is the day in which we find both Christmas and New Year and therefore we can notice an increase in tweets these days .

We can also plot out the average of tweets in hours of day:

```
nycphil['created_at'] = pd.to_datetime(nycphil['created_at'])

plt.xlabel('Hours of the Day')
plt.ylabel('Number of tweet')
plt.hist(nycphil["created_at"].dt.hour, bins=24)
plt.savefig('prova.pdf', bbox_inches = 'tight')
```

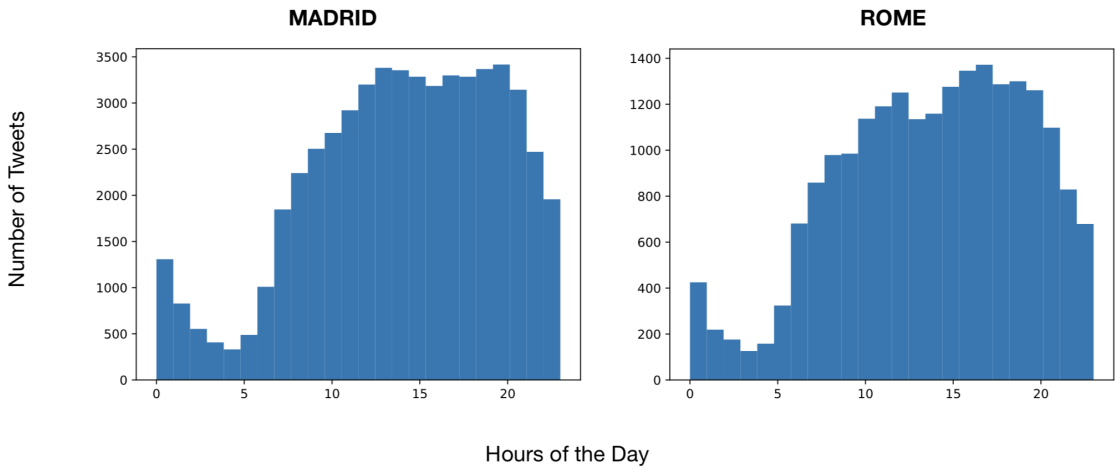

Fig. 6.1. Average number of tweets for hours of the day

Similarly to the previous case, in Fig. 7.1 we can observe that the two graphs have the same trend. It seems that the highest number of tweets is obtained around 1 p.m. and 8 p.m. for Madrid, and around 6 p.m. for Rome, and the lowest number of tweets is obtained in both cases around 5 a.m. This might make sense, as people are sleeping at night so they can't be active on twitter. Conversely, we have the most tweets during lunchtime and when they go out of work. Indeed, these are actually the times when people have more free time and can use social networks.

6.1.2. Subsetting the Tweets

With the aim of doing a more deeper analysis, it is possible to filter the data obtained, reducing considerably the amount of information required to be handled. For example, it is possible to obtain informations that refer only to a certain day or to a short period. In this project we decided to analyze the geographical distribution of the Madrid tweets during the period of New Year's Eve, while we chose the Christmas Day for Rome. To perform this task we used the Folium [45] package, which offers easy features to create interactive maps in python through the use of another library called leaflets [46].

Specifically , and step by step, these were the tasks performed:

- Firstly, filter the values and then create a sub-dataset that contains only tweets that referred to the desired dates.

- Secondly, divide the map into zones according to the distribution of the tweets.
- Finally, for each tweet create a popup in which you can view the attribute "text", ie the text of the tweet itself.

In the Fig. 7.2, Fig. 7.3, Fig. 7.4 there are several examples concerning the results obtained for Madrid during the period from 31/12/2018 to 02/01/2019:

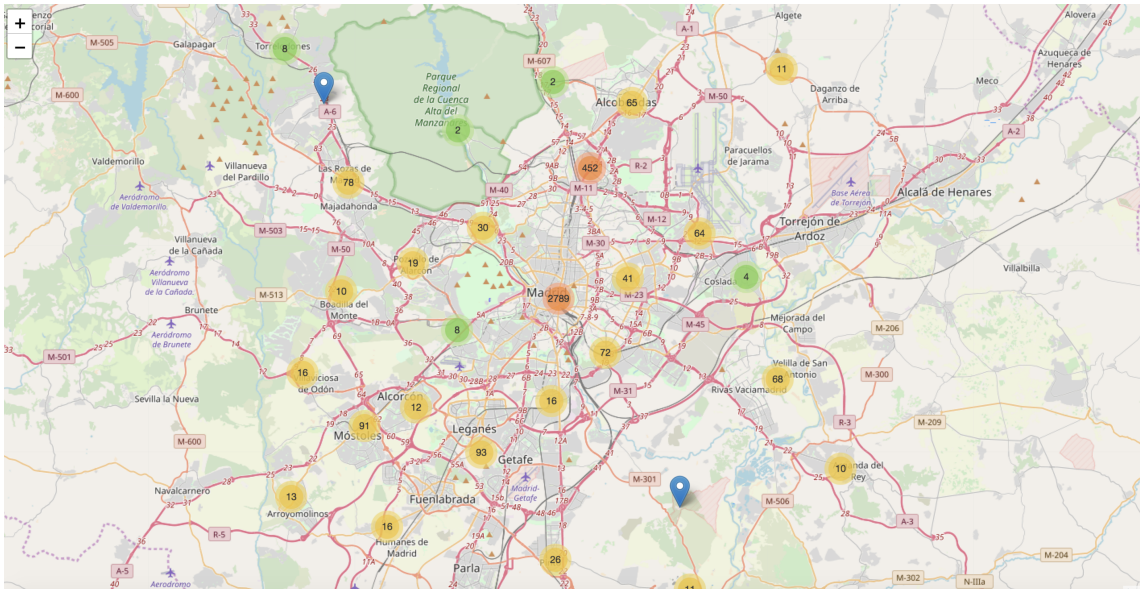

Fig. 6.2. distribution of tweets on the map

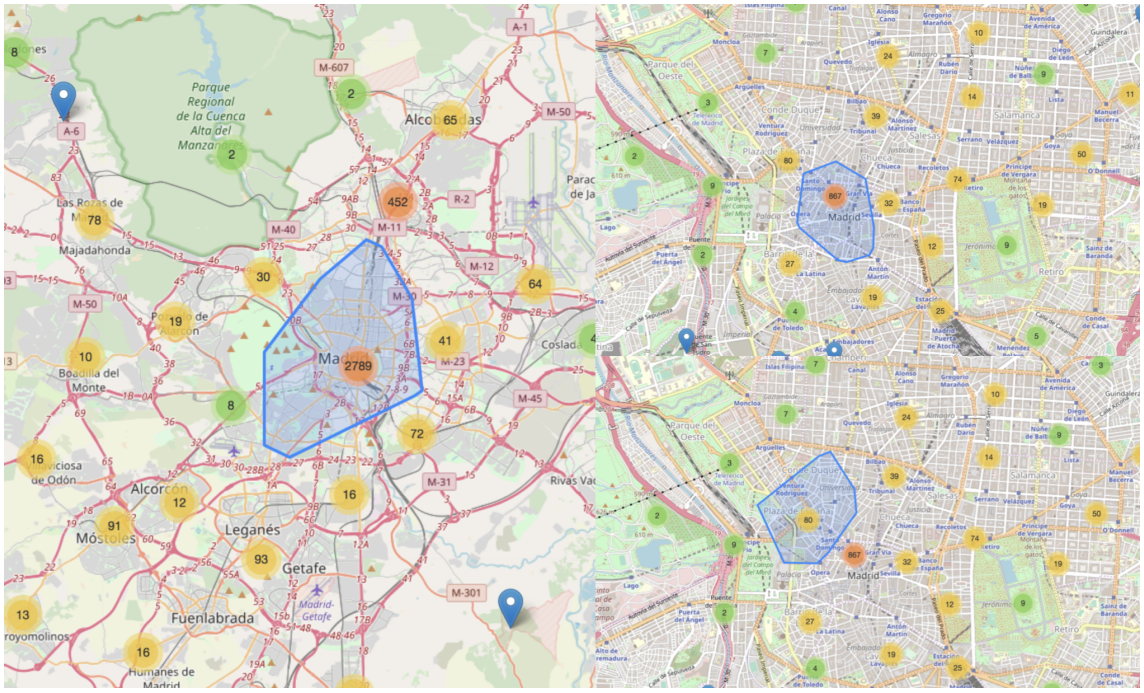

Fig. 6.3. Possibility to choose the area on the map

Fig. 6.4. Viewing the content of a tweet

It is clear how, looking at the Fig. 7.2, the map appears as a whole block. Furthermore there is the possibility to see the number of tweets corresponding to each zone in which the map has been divided. We have more tweets in the central area of Madrid, while as we move away from the center the tweets decrease, possibly due to the density of population, young people and students in these areas. Fig. 7.3 shows the possibility to choose the area to analyze and finally in Fig. 7.4 we see how it is possible to literally 'click' on the tweets and see all the information required. The interesting thing that could be observed is that out of a total number of 4033 tweets posted during the three days in question, 1025 were posted in the central area of the city (more than a quarter of the total number). This figure can be very significant and could be helpful, for instance, for the organization of the city's security.

6.2. Calculation of entropy applied to a specific scenario

In order to calculate the anomalies and compare the two cities taken into consideration, we refer partially to the work and method used in [11]. What we have noticed is that there are parameters that allow us to make evaluations and then find the anomalies. One of these is the period in which the calculation of entropy is performed, that in our case is the Christmas period, i.e period between 10/12/2018 and 10/1/2019. Indeed, we could see anomalies in the days when it was a holiday compared to the same days without holidays. What we have noticed is an increase in the number of tweets in these days and also a higher concentration of tweets in one point of the map compared to another. Another important factor in the calculation is the square size, "N", when you divide the city into a labeled

grid, as this size varies you can get different results.

In the Fig. 7.5 we can see the results obtain with Rome and Madrid coordinates during a Christmas period with the script "without window" showed above and a value on N = 4. It Is possible to notice, according to other tests performed, that the results are different if we change the value of N and using a window that gradually stores the old values we can find out the greatest variations of entropy that allow us to find the real anomalies. The window we have used stores the values starting from 1/3 of the total length of the values; and also in this case, changing this value it is possible to obtain different results, but not with significant variations. Instead, in the Fig. 7.6 we can see the results obtain with the code 7 showed in the previous chapter.

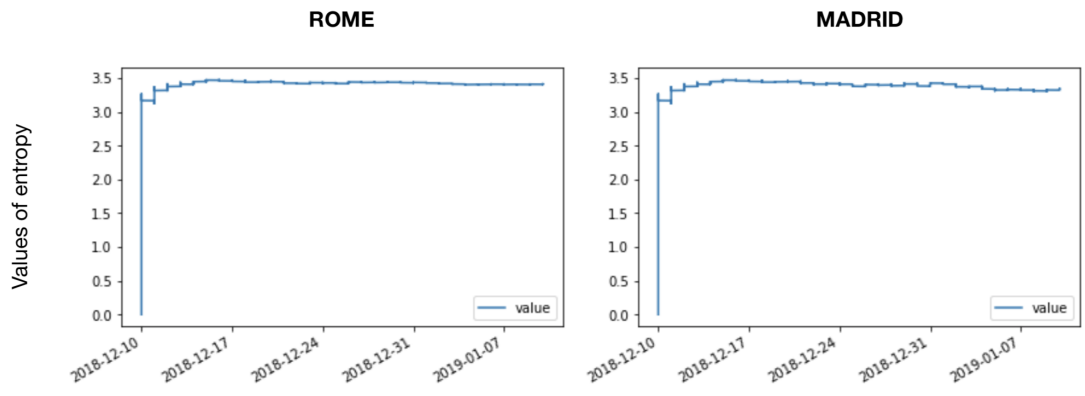

Fig. 6.5. Entropy evolution in the Madrid and Rome data set.

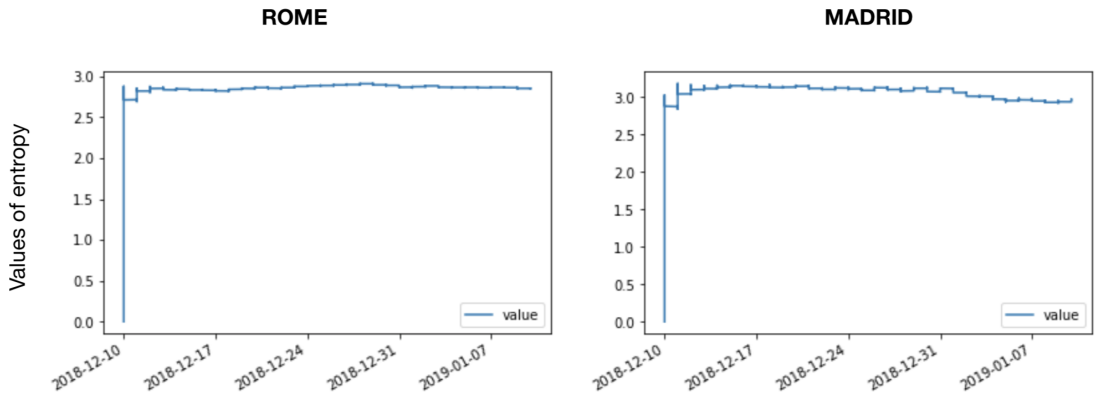

Fig. 6.6. Entropy evolution in the Madrid and Rome data set with window.

What we can immediately notice is the almost total similarity between the graph obtained for Madrid and the one obtained for Rome. This is certainly due to the fact that during the Christmas period the people living in the two capitals have more or less the same

habits. We were able to obtain slightly more significant results for Madrid because the use of this social network is more common with respect to Italian cities, as demonstrated by the magazine "CityMetric" who did an investigation into which city Twitter was most used in [47].

Due to the significant limitations that have been imposed in recent years on the use of these APIs, the data that we have managed to collect concern only one month. For this short period it is not possible to see significant anomalies in the calculation of entropy. This is because it would be appropriate to analyze, for example, the behavior of the crowd on a specific day of the week over a period of at least one year. Nevertheless, by analyzing the entropy of a single day of the week, it is possible to see anomalies on the days when the tweet flow is highest. In our case we have chosen Tuesday as the day, because in the period we have taken into consideration for our experiment, Tuesday is the day on which we have both Christmas and New Year's Eve. In Fig. 7.7 shows the entropy evolution of Tuesdays using the script "with windows" and using the same parameters chosen in the example in the Fig. 7.6.

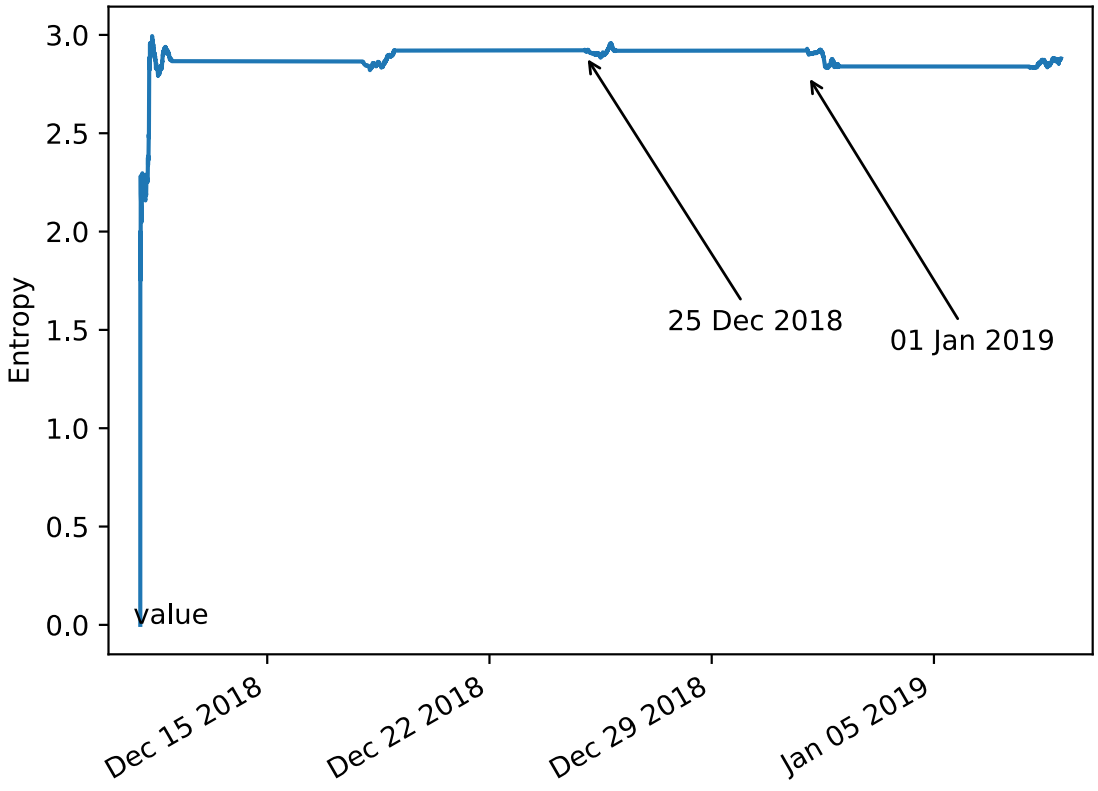

Fig. 6.7. Entropy evolution in the Madrid data set with selected parameters.

7. CONCLUSION

As we have seen, there are many possible methodologies and analyses in the area of social networks and what we have presented until now is an attempt to unify, better define and formalize the most used forms and starting from them, defining new and more rigorous ones, which are able to detail and justify the observations deduced from their results. The project initiated in this thesis is focused on the creation of a method that allows to extrapolate geolocalized data in real time from Twitter and then analyze them in different forms, through the use of an interactive map, several statistics and through the calculation of entropy for the detection of anomalies in urban areas. The knowledge used comes from different fields, such as programming skills in scraping information from pages and websites, the development of graphics and interactive maps and finally the understanding and use of an algorithm for calculating entropy. The intention is to analyze the distribution of tweets in a specific geographical area in order to understand how in a specific period the crowd moves more in one area than in another, to compare the results obtained with an urban area of another city and finally try to identify eventual anomalies. From the results of the tests obtained it is possible to notice several and important information that can be used in many fields of research and data analysis. In fact, the possibility of being able to see the variation of the behaviour of the crowd of one city compared to another in the same period, is a fundamental factor. Moreover, the calculation of entropy, which permits to identify any anomalous events, is even more important. In the future, this kind of analysis could be used to prevent events with a large number of people, such as concerts, celebrations and even assaults. If we consider the problem from a business point of view, the techniques presented represent a complex of methodologies and metrics from which we can obtain data based on the requirements that we want to meet in the analysis and reporting. In this context, a collaborative approach is essential: the company defines requirements, the analysis expert decides which methodologies, metrics and techniques to use and how to set up the report based on a knowledge of the target company. Finally, a third figure is required, that is the expert in the field of data collection. The analysis of social media and reporting of results gives rise to some problems such as understanding requirements, choosing the most appropriate instruments and adapting them, recovering the data needed for analysis, analyzing data and extrapolating the results and the considerations. Last but not least, they should be presented in a formal way as understandable as possible by the requesting company. In the improvement of every single step of this process we find problems still to be analyzed and studied that represent or may represent current and future developments. The main one is the evaluation and improvement of the quality of the data to be analyzed with the application of a selection of the data obtained by rejecting unnecessary files, in order to make the algorithms used work more easily. In general, the area on which we have studied is still not very detailed, but it is a focus of general interest. Companies and brands are showing increasing interest in this type of analysis and in the results that they can bring,

to improve business, communication with customers as well as customer loyalty. This project was born with the idea of providing a simple method to collect data geolocalized by Twitter and be able to analyze them easily. Currently the panorama in this area is really huge and this is just a small drop in the ocean of the future, but I am honored to be part of it.

BIBLIOGRAPHY

[1] F. Fedrigo, "La potenzialità dell'analisi dell'utilizzo dei social network ai fini di marketing", Master thesis in Marketing, Universita Ca' Foscari Venezia, 2014.

[2] M. Russell, *Mining the Social Web: Data Mining Facebook, Twitter, LinkedIn, Google+, GitHub, and More*. O'Reilly Madia, 2018, vol. 3.

[3] A. C. Madrigal, "With instagram's founders out, welcome to facebook inc.", *The Atlantic*, A. LaFrance, Ed., 2018. [Online]. Available: https://www.theatlantic.com/technology/archive/2018/09/with-instagrams-founders-out-welcome-to-facebook-inc/571234/.

[4] J. Tidey, "Social network data: Twitter vs fb vs google vs everyone else", *Towards Data Science*, L. Benistant, Ed., 2017. [Online]. Available: https://towardsdatascience.com/social-network-data-twitter-vs-fb-vs-google-vs-everyone-else-830ea0291c86.

[5] T. Daugherty, M. S. Eastin, and L. Bright, "Exploring consumer motivations for creating user-generated content", *journal of Interactive Advertising*, vol. 8, 2013.

[6] B. Ur, P. G. Leon, L. F. Cranor, R. Shay, and Y. Wang, "Smart, useful, scary, creepy: Perceptions of online behavioral advertising", *Proceedings of the Eighth Symposium on Usable Privacy and Security*, 2012.

[7] E. Vargiu and M. Urru, "Exploiting web scraping in a collaborative filtering- based approach to web advertising", *Artificial Intelligence Research*, 2013. [Online]. Available: https://doi.org/10.5430/air.v2n1p44.

[8] (2016). Customized web data scraping services, [Online]. Available: https://topwebscrapingservice.wordpress.com/category/web-scraping/.

[9] L. Cockcroft and G. Wheeler, "What is entropy?", *educationInChemistry*, 2009. [Online]. Available: https://eic.rsc.org/feature/what-is-entropy/2020274.article.

[10] D. Elia, "Entropia nella teoria dell'informazione", *Scienza per tutti*, 2015. [Online]. Available: http://scienzapertutti.lnf.infn.it/chiedi-allesperto/tutte-le-risposte/2017-0431-entropia-nella-teoria-dell-informazione.

[11] J. F. Silva, "Shannon entropy estimation in -alphabets from convergence results: Studying plug-in estimators", *Electronics*, vol. 1, 2018.

[12] L. Serrano. (2017). Shannon entropy, information gain, and picking balls from buckets, [Online]. Available: https://medium.com/udacity/shannon-entropy-information-gain-and-picking-balls-from-buckets-5810d35d54b4.

[13] A. M. Kaplan, “Social media: Back to the roots and back to the future”, *Systems and Information Technology*, vol. 14, no. 2, 2012. [Online]. Available: https://doi.org/10.1108/13287261211232126.

[14] N. S. Lowell, “Twitter as medium and message”, *Magazine Communications of the ACM*, vol. 54, no. 3, 2018. [Online]. Available: https://cacm.acm.org/magazines/2011/3/105332/fulltext.

[15] B. Coder. (2018). Big data in social networks, [Online]. Available: http://beancoder.com/big-data-in-social-networks/.

[16] C. M., K. C.-Y. Chiua, and M. K. Leeb, “Online social networks: Why do students use facebook?”, *Computers in Human Behavior*, vol. 27, 2011. [Online]. Available: https://doi.org/10.1016/j.chb.2010.07.028.

[17] R. Thomas, “Architectural styles and the design of network-based software architectures.”, *Fielding*, 2000.

[18] Douglas, “The application json media type for javascript object notation (json)”, *Crockford*, 2006. [Online]. Available: http://tools.ietf.org/html/%20rfc5849..

[19] Hammer-Lahav, “The oauth 1.0 protocol.”, *Internet Engineering Task Force*, 2010. [Online]. Available: http://tools.ietf.org/html/%20rfc5849..

[20] H. Dick, “The oauth 2.0 authorization framework.”, *Internet Engineering Task Force*, 2012. [Online]. Available: http://tools.ietf.%20org/html/rfc6749.

[21] F. Ficetola. (2012). Dbpedia e il progetto linked data, [Online]. Available: http://www.francescoficetola.it/author/frnk/page/13/.

[22] (2018). Facebook tools explorer, Facebook, Inc, [Online]. Available: https://developers.facebook.com/tools/explorer.

[23] (2018). Official documentation of facebook api, Facebook, Inc, [Online]. Available: documentation[https://developers.facebook.com/docs/reference/login/extended-profile-properties.

[24] (2018). Permission of facebook api, Facebook, Inc, [Online]. Available: https://developers.facebook.com/docs/reference/login/extended-permissions.

[25] (2018). Official documentation of twitter developers, twitter, Inc, [Online]. Available: https://developers.facebook.com/docs/reference/login/extended-permissions.

[26] (2018). Changelog instagram, Facebook, Inc, [Online]. Available: https://www.instagram.com/developer/changelog/.

[27] (2018). Exended profile proprerties of facebook api, Facebook, Inc, [Online]. Available: https://developers.facebook.com/docs/reference/login/extended-profile-properties.

[28] A. Martin, "Cambridge analytica and facebook: What happened and did the company shift many votes?", *Alphr*, J. Bray, Ed., 2018. [Online]. Available: http://www.alphr.com/politics/1008854/cambridge-analytica-facebook-what-happened.

[29] (2018). Facebook sdk for python, GitHub, Inc, [Online]. Available: https://facebook-sdk.readthedocs.io/en/latest/api.html.

[30] (2018). Review instagram, Facebook, Inc, [Online]. Available: https://www.instagram.com/developer/review/.

[31] (2018). Facebook sdk for python doc, GitHub, Inc, [Online]. Available: https://github.com/bear/python-twitter/wiki.

[32] D. Jones. (2012). Python-twitter-examples, [Online]. Available: https://github.com/ideoforms/python-twitter-examples.

[33] (2018). Premium twitter, twitter, Inc, [Online]. Available: https://developer.twitter.com/en/docs/tweets/search/overview.

[34] C. Garcia-Rubio, R. P. D. Redondo, C. Campo, and A. F. Vilas, "Using entropy of social media location data for agile detection of crowd dynamics anomalies", *Electronics*, 2018.

[35] (2018). Docs twitter, twitter,inc, [Online]. Available: https://developer.twitter.com/en/docs.

[36] (2018). Twitter library, twitter,inc, [Online]. Available: https://developer.twitter.com/en/docs/developer-utilities/twitter-libraries#python.

[37] (2018). Tweepy documentation, GitHub, [Online]. Available: https://tweepy.readthedocs.io/en/v3.5.0/index.html.

[38] M. Bonzanini, "Mining twitter data with python (part 1: Collecting data)", *Python, Data Science, Text Analytics*, 2015. [Online]. Available: https://marcobonzanini.com/2015/03/02/mining-twitter-data-with-python-part-1/.

[39] (2018). Interface place, twitter, Inc, [Online]. Available: http://twitter4j.org/javadoc/twitter4j/Place.html.

[40] Y. Gao, I. Kontoyiannis, and E. Bienenstock, "Estimating the entropy of binary time series: Methodology, some theory and a simulation study", *Entropy*, vol. 10, 2008.

[41] (2019). Panda documentation, GitHub, [Online]. Available: https://pandas.pydata.org/pandas-docs/stable/.

[42] (2018). Response twitter codes, twitter, Inc, [Online]. Available: https://developer.twitter.com/en/docs/basics/response-codes.html.

[43] S. Wu, “Twitter and privacy: Nearly one-in-five tweets divulge user location through geotagging or metadata”, *Press Room*, 2013. [Online]. Available: https://pressroom.usc.edu/twitter-and-privacy-nearly-one-in-five-tweets-divulge-user-location-through-geotagging-or-metadata.

[44] M. Marciniec, “Observing world tweeting tendencies in real-time – part 2”, *Codete*, 2017. [Online]. Available: https://codete.com/blog/observing-world-tweeting-tendencies-in-real-time-part-2/.

[45] (2018). Folium package, github, Inc, [Online]. Available: https://github.com/python-visualization/folium.

[46] (2018). Leaflet library, Vladimir Agafonkin, OpenStreetMap contributors, [Online]. Available: https://leafletjs.com.

[47] B. Speed, “Which city tweets the most?”, *CityMetric*, 2014. [Online]. Available: https://www.citymetric.com/which-city-tweets-most.

[48] (2018). Explorer tools of facebook api, Facebook, Inc, [Online]. Available: https://developers.facebook.com/tools/explorer.

[49] J. A. K. J. B. Schafer and J. Riedl, “E-commerce recommendation applica- tions”, *Applications of data mining to electronic commerce*, 2001.

[50] (2018). Jupyter notebook, Jupiter, Inc, [Online]. Available: https://jupyter.org/about.