Goia Ariana-Carmen

Extracting news and Tweets from the UK about COVID19 in parallel with Lithops and realizing a Sentiment analysis over them and comparing the Opinions.

Master's thesis

Supervised by Marc Sanchez Artigas

University Master's Degree in Computer Security Engineering and Artificial Intelligence

UNIVERSITAT ROVIRA i VIRGILI

Tarragona
2021

Abstract

Nowadays technology has taken over many departments and it's more and more visible in our lives. Society is used to having all their information in their hand with the help of their phone or laptops and that's thanks to this era of technology, knowledge. Everyday people can read millions and millions of news from many different sites, news that has information and information is the key; the best “weapon” at this moment. The things that have read or know from the news, people will have a positive, negative or neutral opinion, but everything depends on the media and how they react.

Last year, the COVID-19 pandemic started, people and media had different points of views. This paper will show how people see the real situation and how the news/media interprets it in a way or another and there will be a sentiment analysis to get the results. By automatically gathering tweets on Twitter from people who reside in the UK and processing them with the help of Lithops [1], e.g., through a parallel sentiment analysis, one can infer how the population “sees” the evolution of the COVID-19 pandemic. Optionally, this view can be complemented with the “official version” from the UK government, e.g., via web scraping, and after performing a sentiment analysis, determine the view of the UK government. Finally, we can cross compare both versions and measure the differences.

Resum

Avui en dia la tecnologia s'ha apoderat de molts departaments i és cada vegada més visible a les nostres vides. La societat està acostumada a tenir tota la informació a la mà amb l'ajut del telèfon o dels ordinadors portàtils i això és gràcies a aquesta era de la tecnologia, la informació. La gent quotidiana pot llegir milions i milions de notícies de molts llocs diferents i la informació és la clau; la millor "arma" en aquest moment. Respecte a les notícies que la gent ha llegit o a les coses que saben, tindran una opinió positiva, negativa o neutral, però tot depèn dels mitjans i de com reaccionen.L'últim any, va començar la pandèmia COVID-19, la gent i els mitjans de comunicació tenen punts de vista diferents.

Aquest document mostrarà com la gent veu la situació real i com les notícies / mitjans la interpreten d'una manera o altra i hi haurà una anàlisi del sentiment per obtenir els resultats. En recollir automàticament tuits a Twitter de persones que resideixen al Regne Unit i processar-los amb l'ajut de Lithops [1], per exemple, mitjançant una anàlisi sentimental paral·lela, es pot inferir com la població "veu" l'evolució de la pandèmia COVID-19 . Opcionalment, aquesta visió es pot complementar amb la "versió oficial" del govern del Regne Unit, per exemple, mitjançant una recopilació automàtica de la informació oficial publicada en les web governamentals i, després de realitzar una anàlisi de sentiment, determinar la visió del govern del Regne Unit. Finalment, podem comparar ambdues versions i mesurar les diferències.

Resumen

Hoy en día la tecnología se ha apoderado de muchos departamentos y es cada vez más visible en nuestras vidas. La sociedad está acostumbrada a tener toda su información en la mano con la ayuda de su teléfono o computadora portátil y eso es gracias a esta era de la tecnología, la información. Todos los días, la gente puede leer millones y millones de noticias de muchos sitios diferentes y la información es la clave; la mejor "arma" en este momento. Respecto a las noticias que la gente ha leído o lo que sabe, tendrá una opinión positiva, negativa o neutral, pero todo depende de los medios y de cómo reaccionen. En el último año, comenzó la pandemia de COVID-19, la gente y los medios tenían diferentes puntos de vista.

Este documento mostrará cómo la gente ve la situación real y cómo las noticias / medios la interpretan de una forma u otra y habrá un análisis de sentimiento para obtener los resultados. Al recopilar automáticamente tweets en Twitter de personas que residen en el Reino Unido y procesarlos con la ayuda de Lithops [1], por ejemplo, a través de un análisis de sentimiento paralelo, se puede inferir cómo la población "ve" la evolución de la pandemia de COVID-19. . Opcionalmente, esta vista se puede complementar con la "versión oficial" del gobierno del Reino Unido, por ejemplo, a través de web scraping y, después de realizar un análisis de sentimiento, determinar la opinión del gobierno del Reino Unido. Finalmente, podemos comparar ambas versiones y medir las diferencias.

Contents:

Acknowledgements

I would like to thank my supervisor Marc Sanchez Artigas who has been helping me and providing me invaluable advice, continuous support, and patience during my study. I would also like to thank my family and friends for supporting me, helping me with proof reading this document and believing in me during this hard time.

1. Introduction

Last year, the world was assaulted with the same news all around the globe: SARS-CoV-2, a new coronavirus strain, appeared in Wuhan, China in December 2019. On the 11th of March, year 2020, WHO, the World Health Organization, announced that the COVID-19 situation has deteriorated to the point of becoming a global pandemic. This new virus is currently responsible for the death of almost 4 million people, having infected more than 174 million people. The most known symptoms of COVID-19 are the loss of taste, called ageusia, loss of smell, called anosmia, fever, dry cough and tiredness, but it can also cause complications such as pneumonia, shortness of breath and chest pain, which may lead to the death of the patient. To help slow down the spread of the virus, multiple countries resorted to total lockdown that some places lasted for months, which made not only the global economy to degrade considerably, but also the tourism, sanitary and educational systems. Soon, a great majority of people had to adapt to the online system that was implemented almost everywhere. Some of the luckiest people continued their studies and work from home, while others got unemployed or couldn't continue attending classes because of poverty and lack of necessary equipment.

Not only did the economy and different systems of the world get affected, but also people suffered psychologically. The lack of social interactions and sudden changes made by the governments in such a short amount of time sparked doubt in people, who proceeded to ignore the implemented restrictions and even started protests or riots, claiming that the news that they saw about the pandemic is just a hoax and the governments are trying to brainwash them or enslave them. On the other hand, other people, like youtube celebrities, decided to start creating more interactive content, with the purpose of helping their fans and viewers keep a somewhat stable mental condition by making more live streams or simply by creating more content to keep us entertained.

Another serious matter that appeared during the pandemic was the amount of fake news on both the internet and television. Multiple countries showed to the world less cases than there truly were and were claiming that the situation was not as bad as it was anticipated, to keep a somewhat calm approach to the matter: "stay calm, but keep respecting the restrictions". But soon, more countries started counting every deceased person as a covid case, even if the person died of cancer, heart attack or other non related causes, making the numbers go up a lot, some european countries even adopting this method, only to get funds from the EU.

As months went by, people's mental health started deteriorating considerably because of the constant fake news, deaths, revolts, protests, riots and overall, the lack of communication between one another, which led people taking refuge and venting to others, especially on twitter. Venting was like a salvation to most people, helping them release all the pressure that accumulated over time. Using people's determination to keep in check with the world, we can create statistics which can help in analysing the different points of views, as well as the government responses.

1.1 Objectives

This work is dedicated to build a programmatic workflow for the automatic analysis of the COVID-19 pandemic on people from the UK and the news that the government of the UK wants to share. In this work, the goal is divided in six objectives:

1. Search for reliable sites to gather relevant information about COVID-19 in the UK
2. Retrieval of tweets related to COVID-19
3. Analyze the data gathered
4. Use Lithops to store the big flow of data from Twitter
5. Start a sentiment analysis on both data sets
6. Draw some conclusions

1.2 Document Structure

This document will be organized as follows:

Chapter 2: State of the art

It will be a concise state of the art on the various technologies we utilized in this project. From serverless computing where we focus on the library we used - Lithops, to Sentiment analysis and web scraping.

Chapter 3: Describing the project

Here we will describe how the program was created, step-by-step, focusing on every aspect: from gathering the data sets that includes the society view - Twitter and the government view - government news site, the sentiment analysis that was made on those data sets and finally how we compare the results.

Chapter 4: Experimental results

Here we will show all the experiments we did, what the graphics show and the final result.

Chapter 5: Conclusion

Here we will write our conclusion and our point of view on how this project can help and what the results really mean for us.

Lastly, at the end of the document, there will be a list of all the references.

2. State-of-the art

In this chapter there is a brief overview of the state-of-the art studies that are related to this work. There are three main concepts that are used in this work: serverless computing (we use Lithops), Sentiment Analysis and Web Scraping. All of those will be discussed separately and will contain details on how we applied it on the project.

2.1. Serverless computing

The IT industry has experienced considerable innovation during the previous decade: we need more “power” for the programs that are more complex, we need more space to store the data, we need more people to code the functions that we need and so on. With time, data started being the main focus, being so important, we started to store it and they were only byte size but now data became so large and big that we store it in Terabytes or Petabytes.

Not all companies or people have resources in order to store a big flow of information so it started as a need to have a bigger storage that you can access it from any point of the globe whenever you want. This is the moment when Serverless computing was born. The serverless computing revolution has enabled IT businesses to design and deploy scalable applications without worrying about the underlying infrastructure.[2] The fundamental goal of this technical advancement has been to increase company agility, improve resiliency, and create cost reductions. First, there was the virtual machines, then containers and now serverless.

The idea of serverless computing and its numerous services is not formally defined. A FaaS and BaaS combo is a serverless service.

Function as a service (FaaS) is a paradigm, in which clients are capable, without the effort of creating and maintaining infrastructure, to design, run and manage application capabilities.

Backend as a service (BaaS) is an online service that sends a specified task through the cloud. Like FaaS, BaaS does not need customers' resource management. BaaS also offers a complete online service.[3] It is also a cloud-based distributed NoSQL database that eliminates database administration overheads.

Serverless computing was not a new concept for Amazon [4], they debuted their cloud storage service (AWS S3) in 2006, which was also

a 'serverless service' that delivered unlimited storage at infinite scale without the need for servers to be maintained.

By removing the need for developers to manage infrastructure, serverless computing allows them to construct apps more quickly. The cloud service provider automatically provisions, scales, and manages the infrastructure necessary to run the code using serverless apps.

The main benefits of the serverless computing are:

1. Fully managed service, developers focus on the code, not on the infrastructure.
2. Supports event-driven approach
3. Provides infinite scalability and built-in high availability, to meet the needs of any workload, the infrastructure dynamically scales up and down in seconds, depending on the traffic.
4. Less-Ops
5. Pay for execution time, if the application is not running, not used, then you don't have to pay for it, only when it's executed.
6. Faster time to market, all the applications you create, they will be deployed and run in a matter of seconds.

At the same time, there are some disadvantages:

1. Monitoring and debugging are still challenging. The complexity of the management of such situations has only been deliberately amplified. [5]
2. It has cold starts, many times it needs to start up from zero to serve a new request or thanks to the scalability, they have a hard time to start once they reach the lowest point.
3. Not all the applications need the serverless option and isn't the best one. There might be a better option if the workloads are predictable, steady or long-running processes. When provisioned for the expected demand, VM deploys are many times cheaper. [6]
4. There still exist security issues, they’re not more secure than traditional architectures. [7] DDoS attacks or SQL injection attacks still present threats. To raise the chances to avoid this kind of attacks the inputs must be validated, the libraries updated and best practices used. Many serverless operations engage additional data processing applications that provide data storage and recovery. Developers need a reliable framework for managing all the data. [8]
5. Privacy issues, many servers are built on proprietary public cloud infrastructures.

6. Vendor lock-in

The majority of software is designed to operate on a single system, and parallelization may be restricted to the number of computer cores or threads accessible locally. [9] Parallel processing is a typical computer methodology used to handle huge problems in less time. The overall length of a task is considerably decreased by performing the components of the issue simultaneously. [10]

Thanks to the advantages, serverless computing enables industry incumbents to compete in their sector with the new high-tech customers. Being so “wide”, serverless computing can be used in many industries and are prefered than the traditional methods. It's popular when it's about Data processing, parallel computing, map operations, stream processing workloads.

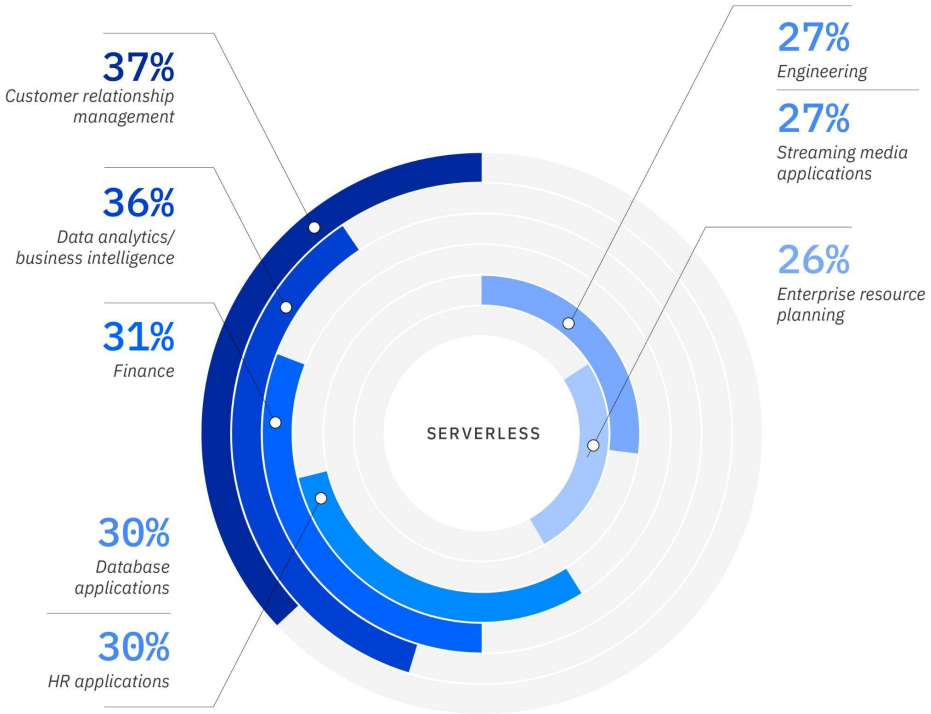

Fig1. in what domains the serverless technology is used

For this work, we will use serverless computing because we need a parallelization thanks to the great flow of data that we will gather and for that, we will use Lithops.

2.1.1. Lithops

Lithops is a Python multi-cloud distributed computing framework. It can be described as a dynamic job orchestrator, designed to run

functions into a serverless computing platform. It can run python code without modifying it in a serverless mode. Lithops erases the disadvantage of the vendor lock in and the user isn't concerned about how the code is deployed or run in the cloud. Lithops is used mostly in problems with big data analytics and embarrassingly parallel jobs. [11]

Lithops has three ways of execution: localhost mode, serverless mode and standalone mode. The localhost mode is the mode where the functions are run in your own machine, no clouds, just processes from the machine. Serverless mode is the mode where the functions are run in accessible serverless computational services like IBM Cloud Functions, Amazon Lambda/S3, Google Cloud Functions, RedHat Openshift and much more. Using serverless mode, the functions will be treated like a parallel task. Lastly, standalone mode is the mode where the functions are run in virtual machines. [12]

Lithops has High-level Compute and Storage APIs: Futures API, Multiprocessing API, Storage API and Storage OS API. We will use the Storage OS API because we will store all the big flow of data in the cloud thanks to Lithops and from the Futures API to run the functions in an asynchronous way. In Futures API, we use map() and call_async() calls.

Lithops provides the ability to generate a Storage instance while abstracting away the underlying implementation of the backend. We will use storage.put_object() because it adds an object (our data) to a bucket of the storage backend. We will need it when it comes to the big data flow that we will gather. Once the object is put in the bucket, we will need to get it and work with it so we will also use the function storage.get_object() which will retrieve objects from the storage backend. The Storage API gives an unprecedented flexibility to run tasks to the Cloud like big data analytics or other kinds of applications that involve data analysis or management.[11] We also use asynchronous functions (we don't need synchronization, the data gathered doesn't need to be read in the same way as how it was sent or vice versa). The functions are: call_async() is a method used to spawn one function activation and the map() method used to spawn multiple function activations. [13] Lithops has an API that is similar to Python’s library of concurrent.futures. The objects Futures are created once Lithops creates a function and thanks to this, the user can access the outcome, as well as various statistics. For example, we may utilize the call async() API method to launch only one function and then manage the resulting Future.

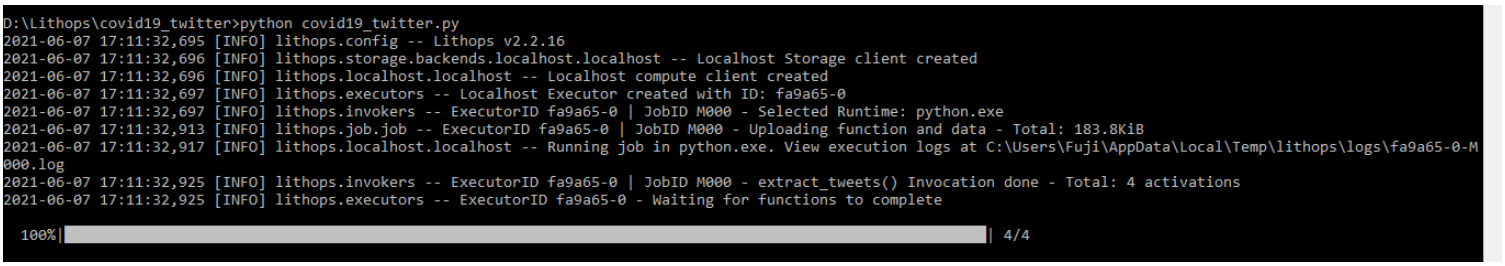

Fig 2. Lithops starting and working on the asynchronous calls

2.2. Sentiment Analysis

Sentiment Analysis is a special Natural Language that helps to determine in a computed way if a text is positive, negative or neutral. It is used widely in many departments: business (if a product is doing well, if people like the product or not, it can influence a lot of a business strategy and the product), politics, publications, public events and so on. [14] It can help to improve client experience, reducing employee turnover, creating products, deleting posts, product analysis, social media monitoring, brand perception, emotion recognition, chatbot reactions and the list continues.

Humans are highly subjective beings, and their viewpoints are valued. The ability to communicate with humans on that level offers several benefits for information systems. [15] People communicate their thoughts in a variety of ways. Lexical content alone can be deceptive in opinion writing.[16] Sentiment analysis is used to discover the user's thoughts on a specific entity and the information that was gathered can be utilized to increase profits and services. Through the analysis process of many text mining viewpoints we can come to an idea and maybe a solution.[17].

There are four types of sentimental analysis: fine grained sentiment, intent analysis, aspect based and emotion detection. Fine grained sentiment gives an understanding of customer feedback, it's really precise. Intent analysis goes deeper, it can recognize what kind of text it is: complaint, suggestion, recommendation, question opinions, sentiments, evaluations, appraisals, attitudes, and so on. Aspect based focuses on only one thing that we want to find out, for

example if we want to know how the product is working, not the package, not the problems we use this kind of analysis. Lastly, emotion detection, how the name states, it detects the emotion behind the text, if the person was sad while they wrote the post, happy, angry and much more. [18]

Sentiment analysis is a challenging process because of many reasons, one of them being the fact that people can write in a sarcastic way and the sentiment analysis might not get the main idea. The difficulty grows in proportion to the richness and complexity of the viewpoints conveyed. Product reviews, for example, are quite simple for sentiment analysis but books, movies, art, and music are more difficult to understand. Policy conversations are more challenging, and indirect statements of opinion are much more challenging. [19]

According to more surveys, we can see that individuals and organizations are increasingly relying on the content of various media to make decisions.[20] [21][22]:

- 93% of customers read online reviews before buying a product.
- 91% of 18-34 year olds trust online reviews as much as personal recommendations
- Nearly all consumers (97%) now use online media when researching products or services in their local area.
- Positive reviews remain a key way for companies to sell their product, with customers willing to spend 31% more on a business with excellent reviews.
- 92% of buyers are more likely to purchase after reading a trusted review.
- 3.3 is the minimum star rating of a business consumers would engage with and only 13% of consumers will consider using a business that has a 1 or 2 star rating
- 94% say an online review has convinced them to avoid a business.
- Four out of five consumers have changed their minds about a recommended purchase after reading negative online reviews.
- About 95% of customers read reviews before making a purchase.
- 93% of customers will read reviews of local businesses to determine its quality.
- 72% of customers won't take any buying actions until they've read reviews.
- When a product gets five reviews, the likelihood of it being purchased increases by 270%.

- 82% of customers actively seek out negative reviews.
- Negative reviews can stop an average of 40% of buyers from wanting to buy from a business.
- Businesses whose total number of reviews are 15-20% negative actually average 13% more revenue than businesses whose total number of reviews are 5-10% negative.
- 72% of buyers say negative reviews give depth and insight into a product.
- 40% of buyers say negative reviews help build credibility for a product.
- 95% of customers get suspicious of a rating if there are no negative reviews.
- The likelihood of purchase peaks at a star rating of 4.0 to 4.7, then decreases as the rating gets closer to 5.0.

We can see that people are interested in the opinions of others, more if they have something good or bad to say. We will, in the same way, analyze people's opinions regarding COVID-19 and the measures by the government of the UK and the news that they report. We are looking after how they feel about this topic: in a positive way, negative way or a neutral way. The expression of emotion is nuanced, blended, and continuous (Russell 1980; Ekman 1992; Wilson et al. 2006): Human reactions are complicated and multi-dimensional as well, insisting on a single label fails to honor the author's objectives and results in untrustworthy labels.[23] But even so, the sentiment analysis that we will do on the tweets will have a meaning and a contribution to this work.

To do the sentimental analysis on this project, we will be using Textblob and with its analysis we will know the sentiment (positive, negative or neutral) and the polarities that define the sentiment. The polarity works by extracting "opinion sentences" based on the presence of a predefined set of product attributes and adjectives. After that, it will analyze the sentences based on the number of positive polarity words vs negative polarity terms. TextBlob is a Python library with an easy-to-use API for accessing its functions and doing basic NLP operations. The best feature of TextBlob is that it works like lists in python, things that can help us in our work and be easier to manipulate the data.

For a given input sentence, the Textblob sentiment analyzer provides two properties:

Polarity is a float that ranges between [-1,1], where the minimum represents negative emotion and the maximum represents positive emotion. The middle, zero, will be the neutral polarity.

Subjectivity is also a float with a value between [0,1]. Subjective sentences are used to express one's opinion, feelings, or judgment.

For this project, we will just need the polarity.

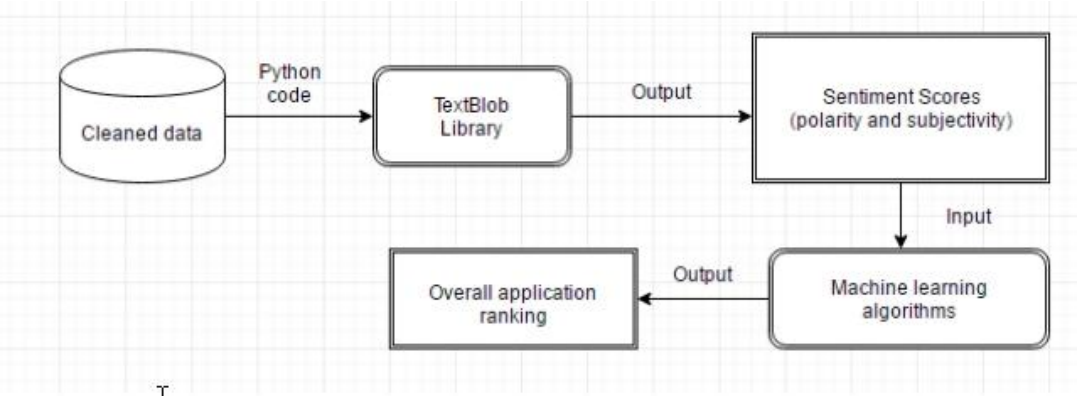

Fig 3. How TextBlob works

2.3. Web scraping

Web scraping, also known as web data mining or web harvesting, is to write a program that can automatically collect, interpret, retrieve, and arrange meaningful information from the web. Web scraping is a short-cut, because instead of manually storing data from websites, it will automatically load and extract data from various websites based on our specifications. [24]

The applications for utilizing web scraping are as many as the applications and reasons for utilizing the Internet. Web scrapers may perform anything, such as ordering online meals, buying in the middle of the night concert tickets before they're sold out, among other things, just like a human being is capable of. Some of the most important applications of web scraping are: collecting data for Machine Learning Projects, data for research, in Search Engine Optimization (SEO), travel industry, banks, e-commerce, journalism and so on.[25]

There are many tools that can help with web scraping [26] and aren't required to start your own web scraping tool, also APIs are sometimes provided by websites or online services to get or interact with data. It is not unusual though that APIs are unavailable or that the solutions provided do not fulfill user requirements.[27] Building an API can be expensive for businesses (they have to create it, test it, keep the updates and versions, create documentation on their expense) and some businesses don't bother with creating them., and there are infrastructure

and engineering costs to consider. Moreover, maybe the API's have limits, for example, 100 tweets per day, 100 news per day and it limits the data and everything that you want to do.

Web scraping is built up in three stages:

Fetching stage - We need the url of the site that we want to web scrape and it needs to be accessed. This is accomplished using the HTTP protocol, which is an Internet protocol used to send requests and receive replies from a web server. Web browsers employ similar mechanisms to obtain web page content.

We use Requests from Python for sending a HTTP GET request to the requested site (URL) and receiving a response, if the site is responding 200 then it means that the request has succeeded. After getting a positive response, we use BeautifulSoup to extract the HTML page.

Extraction stage- After retrieving the HTML page, the relevant data must be extracted. Regular expressions, HTML parsing libraries, or XPath queries are employed. For the extraction, we will use BeautifulSoup. This library has all the functions we need to get all the data we need to gather.

Transformation stage- Now that just the relevant data is left, it may be turned into a structured format for storage or presentation. [28] We will be using data frames, csv and lists to save all the information we gathered.

BeautifulSoup is a Python library for extracting HTML. It's simple, easy for navigating, searching, and modifying parse trees, such as an HTML tree. It is designed to be simple to use and includes traversal features such as discovering all links or all tables that match a certain criterion. [29] Nearly any HTML (or XML) file might be retrieved as long as it contains an identifying tag around it. [30] After using BeautifulSoup to parse the HTML, now we want to extract some valuable information from the HTML text. The soup object includes all of the data from the hierarchical structure that may be retrieved programmatically. We will use the functions find() and find_all(), the first one returns the first matching element and the other returns all the matching elements.

3. Describing the project

The main goal of this work is to create a program that gathers all the information needed about a subject from Twitter and from a certain site. After we have the data, we clean it and analyze it so we can get the sentiment from our results . Once we have the sentiments, we can compare the views, finally getting a final result. In this work, we will focus on news and tweets related to COVID-19 from the UK. The tweets will be from all around the UK and the news will be recovered from the official government site from the UK, gov.uk.

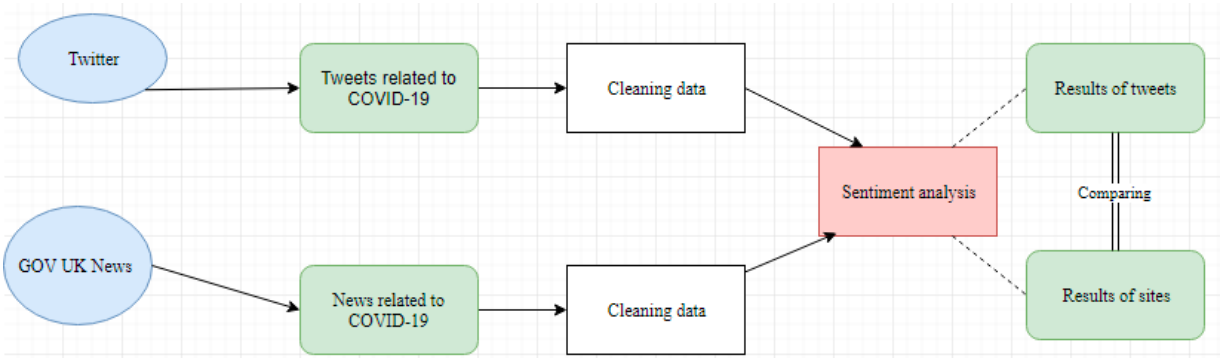

Fig 4. How the program works

We picked Twitter because there was an unlimited number of tweets (publications) that users can create, most of them being public. On Twitter, people can share their thoughts, but in order to make a comparison we need their opinion. In comparison with other big social media platforms, Facebook requires you to access someone's profile to see their posts (if they're public) and Instagram consists of photos.

The gov.uk site was chosen for many reasons but the most important ones are that the news are from a reliable site that isn't biased, the information has to go through many filters and only after that, the news is approved and therefore published. Being from the government, it gives a sense of trust, the numbers that they give will not be modified in their benefit and they will be the real ones. Many sites tend to create news without research or looking deeply into the source of the information or others just create fake news for their own interests. On this site, it is guaranteed that there will be only trustworthy news with verified sources and this is helping us because we need a truthful sentiment analysis based on true stories.

3.1. Gathering data sets

To create the comparison, we need two data sets, one representing the data set for the society and one representing the government. The data set from the society will be tweets from Twitter, of people who reside in the UK. The data set from the government will be the news from the government site of the UK.

Those two, will be in different sections because there was a different approach for each one to gather the information needed.

3.1.1. Society view: Twitter

To gather tweets from Twitter, we need the API they provided and it can only be used if you have a developer account for Twitter[31]. The API provided in Twitter is specially made for retrieving and analyzing the users, tweets, lists, trends, media and places. It has all those resources to choose from, but we are only interested in the tweets that are related to COVID-19 and their location (they need to be posted from the UK). The API will let us look after tweets and gather them in a place and to do that we need a standard developer account[32]. After taking actions to get the developer account, we could freely use the API and the functions that includes.

Once we have the account, we create a project and an app to generate credentials: a consumer key (username, used by the application to make a request), consumer secret (password, used by the application to make a request), an access token and access token secret (both representing the Twitter account that owns the App). All of those keys and secrets are needed to create the requests and to retrieve the “solution”. After having the keys and secrets, we create an authentication and finally we can use the API only if the credentials are valid.

First, we have to think what kind of words we want to look after in the tweets. Because we want to focus on the COVID-19 pandemic, we will look after the words ”#COVID-19”, “covid-19”, “Pandemic”, “Lockdown”, “covid19”, “COVID19”, “coronavirus”, “corona” and all the variations related to them (with or without capitalized first letter or in all caps). All the words are related to the COVID-19 and those words will be the “search_words”.

Secondly, we need to think about cleaning the text from the tweets: the tweets might have emojis (emoticons, symbols, flags and much more), links and structures of words/data that aren't related to the tweet but they appear there. We have to erase all the text that isn't relevant to our experiment. If we don't erase them, those words can't be used in the sentiment analysis (it doesn't have any meaning so they aren't relevant), it doesn't have any polarity and also if they appear many times, they will appear in a word cloud alongside with the words that are relevant. All the cleaning is on a function named "data_cleaning".

Now to gather the tweets, we use the function "extract_tweets". Here, thanks to the Twitter's API, we create a Cursor for the pagination and we send the request to search tweets and returns a collection of relevant Tweets matching a specified query, the query being the search words we are looking after. To guarantee that the tweets are in english, we add the language to be english. To be sure that the tweets are from the UK, we use the parameter geocode that returns tweets by users located within a given radius of the given latitude/longitude. We used this parameter because it's more reliable than relying on users that wrote in their location UK but in reality, they're in another country. To use geocode, we look after the coordinates of the center of the UK (latitude and longitude) and then we pick a radius that will include the UK territory.

```
#function that extracts tweets of Twitter
def extract_tweets(id, bucket_name, key, n_iter, storage):
    import pandas as pd

    tweets=tw.Cursor(api.search,
                     q=search_words,
                     geocode= "52.633331,-1.133333,300km",
                     lang="en",
                     since=startDate,
                     until=endDate).items(n_iter)
```

Fig 5. Code from the function "extract_tweets", it shows how the tweets are extracted.

After getting the tweets, we will check if the tweets aren't retweets, in other words they are not duplicates or they are not replies to a tweet. Having replicas in the data sets, the sentiment of those will grow exponentially, even though we are collecting the same data over and over again. For example: we gather a tweet that states "I like my car" that has 5 other duplicates (retweets) and another tweet that states "I don't like my car". In this case, we would receive a 6 to 1 ratio with

6 positive sentiments and one negative sentiment, even though there are originally only 2 different tweets. Furthermore, if we eliminate the replicas, we would come back to the 1 to 1 ratio, one positive and one negative, which is the correct solution and it shows the real situation.

Finally, once we have the tweets, we only extract the text and the location, ignoring the rest of data that has the tweet. Having those two, we use Lithops to save the data in parallel, serverless, in a bucket that provides Lithop.

```
            tuple=(
                    tweet.text.encode('utf-8'),
                    tweet.user.location.encode('utf-8')
                )
            twl.append(tuple)
    df=pd.DataFrame(twl, columns=['text', 'location'])
    storage.put_object(bucket_name, key+'{}.df'.format(id), pickle.dumps(df))
```

Fig 6. Code from the function “extract_tweets”, it shows how the tweets are stored with the help of Lithops.

3.1.2. Government view: government news site

To gather information on a site, we need to start a process of web scraping. There are many applications already made that do not need the process of writing your own code and just use the user-friendly application to gather your own information. Finding the best program for our site will take a lot of time and it will need to be personalized because of one reason: we need the articles that are posted from a specific day. We reached a conclusion: we will write our own web scraping tool personalized for our site, it will be easier to modify, easier to work with and really simple. Moreover, we need the information that we gather to be analyzed and finally to be put in comparison with the other data set that we have.

First, we have the site from where we will take all the information, the government site of the UK. We filtered the subject only to be related to COVID-19, there will be all the news related to our subject and there is no more need to make a filter in our program.

After having the filtered list of news, we will look at the number of pages, this being important information because the program might have to go through all the pages to find the news from a specific day. Using requests, it makes a request to a web page, and returns the status code (200 - meaning the request has succeeded). We will use the library Beautifulsoup, which makes it easy to scrape information

from web pages [33] and with the help of it, we can do our own web scraping.

Manually, we will go to our site and we will click on “Inspect Elements” so we can know the tags and classes we need to realize the web scraping. The site has many elements, lists and so on but we only need, from this link, the links that represent the articles. We can see that there is a list of articles and we need the links of the articles but also we need the time posted. Time is crucial to us because we will compare the sentiments of the articles from day X with the sentiments of the tweets of the same day X. Comparing sentiments of different days will not give a reliable result so we need to be careful about selecting the same day. We found the class “gem-c-document-list gem-c-document-list--no-underline” and we saw that it includes the links we needed and the time when it was published. Once we find the links to the articles we need, we save it into a list.

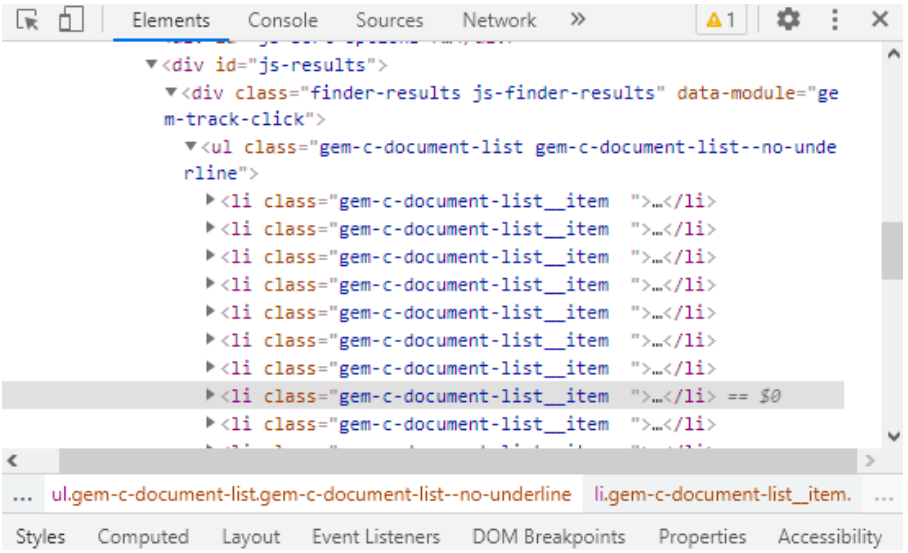

Fig 7. How inspect element looks and how did we find the class we need

Having the list of the links of the articles, now we only need to go through all the links and take the information of the article. To do that, we still need to take the first steps we made before: make a request, use Beautifulsoup, use the “Inspect element” feature and check for the tags and classes that we need. We see that the articles have all their information in paragraphs (using the <p> tag) and we will save separately, every paragraph in a csv document.

```
req = requests.get(url + li[i]+'/') #make a request to check if the link is available
soup = bs(req.text, 'html.parser') #parsing the site
par = soup.find(class_='gem-c-govspeak govuk-govspeak direction-ltr') #finding again the class that cointains the information
if(par.find_all('p')!=None):
    conti = par.find_all('p')
```

Fig.8. Piece of code that realizes the request and uses the Beautifulsoup functions to web scrape.

We saved separately because of one important reason, a document can have, for example, 10 paragraphs and if we take it as a whole, the sentiment analysis will be a sentiment of the whole article but in an erroneous way, it will be the median sentiment. Continuing with the example, from 10 paragraphs, 7 paragraphs have a positive sentiment, 2 are negative and one is neutral. The sentiment analysis will be done to all the paragraphs and it will be a positive one only because there are more positive paragraphs than negatives/neutral ones. The csv document will be a document with paragraphs of all the articles that we found from the site.

After gathering all the data sets needed, now we can go to the next part, the sentiment analysis.

3.2 Sentiment analysis

The sentiment analysis is used in the same way for both data sets. We will create one function that will classify sentiment of a passed tweet or paragraph using Textblob's sentiment method. This method has a polarity score that is a float within the range [-1.0, 1.0] [34], if the polarity is bigger than 0, then is a positive sentiment, if it's equal to 0, then is a neutral sentiment and lastly, if it's smaller than 0 then is a negative sentiment.

```
#Utility function to classify sentiment of passed paragraph using textblob's sentiment method
def get_tweet_sentiment(tx):
        analysis = TextBlob(tx)# create TextBlob object of passed tweet text
        # set sentiment
        if analysis.sentiment.polarity > 0:
            return 'positive'
        elif analysis.sentiment.polarity == 0:
            return 'neutral'
        else:
            return 'negative'
```

Fig.9. Code of the sentimental analysis

For the data sets, we check the polarity and return the sentiment we got. At the same time, we also need the polarities and that's because the numbers that we will get, will be shared at the end of the

comparison. After getting the sentiments of every data set, we can now compare them.

3.3 Comparing the results

In this section, we will compare the results of the tweets and news of the day of 02-June-2021. To compare the results, we decided to first print the sentiment and the polarity of every tweet gathered then it will be shown a bar plot showing how many positive, neutral and negative sentiments there are. Then, we will do the same for the government. Each one will have their own word cloud, it will help us to show which words are the most used, giving us an overview on what people are interested in, on what they are concentrated/focused on. Lastly, in the next chapter, Experimental results, we will use boxplots to compare the views of both of them from consecutive days. This will give us a better view on the differences and similarities that they have.

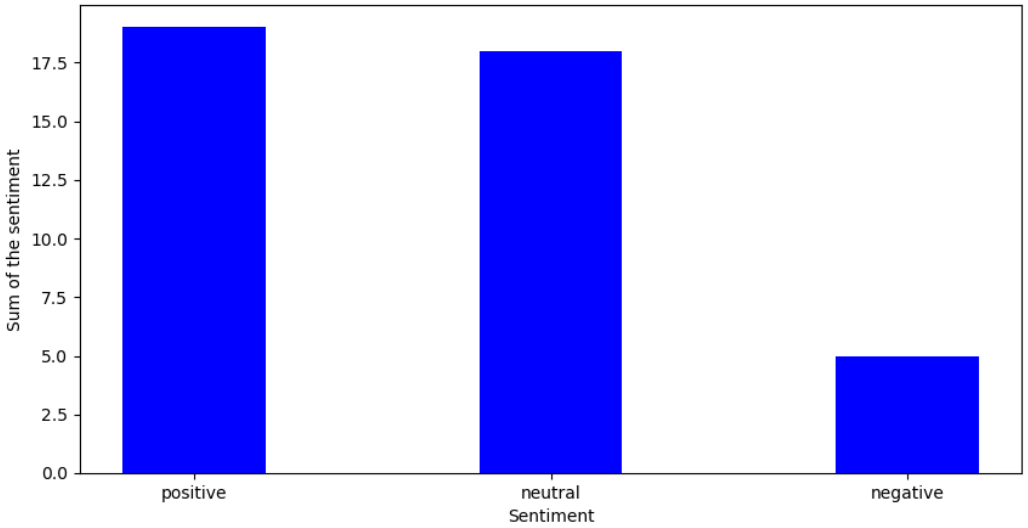

Fig.10 The bar plot resulted with the sentiment gathered from Twitter on 2 June 2021.

0	positive	0.225
1	neutral	0
2	neutral	0
3	neutral	0
4	neutral	0

5	positive	0.25
6	positive	0.170455
7	positive	0.128788
8	positive	0.125
9	positive	0.136364
10	negative	-0.1
11	neutral	0
12	positive	0.5
13	positive	0.068182
14	positive	0.1375
15	positive	0.5
16	negative	-0.18056
17	neutral	0
18	neutral	0
19	negative	-0.5

Fig 11. example of data received from the sentiment analysis

We can see that the predominant sentiments are the positive ones and the neutral ones but there still exists some negative feelings.

Fig 12. The results of the word cloud on 2-June-2021 for the tweets.

We can see in the word clouds that people are still talking about COVID-19, there are some hobbies (dance, book, science) mentioned, the vaccination process, lockdown, webinars that are still a COVID-19 related.

We can see that people are preoccupied but on a positive note or in a neutral note. On 2-June people are already accustomed with their life and COVID-19 aftermath.

```
# function to create a word cloud
def wordcloud(df):
    stopwords = nltk.corpus.stopwords.words('english')
    clean=[]

    for i in df['News']:
        clean_text = re.sub('[^a-zA-Z]+',' ',i)
        clean_text = clean_text.lower()
        clean.append(clean_text)
    # we clean the text so there arent repeated words in the cloud
    text = " ".join(review for review in clean)
    print ("There are {} words in the combination of all review.".format(len(text)))
    wordcloud = WordCloud(stopwords=stopwords, background_color="white").generate(text)

    plt.imshow(wordcloud, interpolation='bilinear')
    plt.axis("off")
    plt.show()
    wordcloud.to_file("img/first_review.png")
```

Fig 12. The code of the word cloud.

Now we will look at the results of the government part:

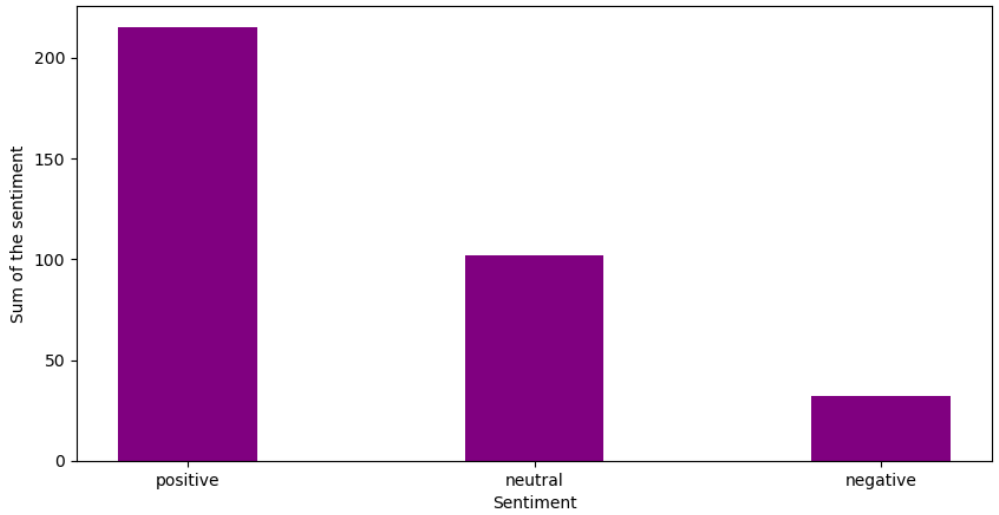

Fig 13. The bar plot resulted with the sentiment gathered from the news on 2 June 2021.

We can see how the articles are written in a more positive manner, even though there are some paragraphs with a negative connotation, there are not as many. The neutral sentiment is still present in some of the paragraphs. There are not as many neutral sentiments as there were for the tweets.

0	positive	0.375
1	positive	0.0625
2	positive	0.183333
3	positive	0.083333
4	positive	0.033333
5	neutral	0
6	positive	0.113636
7	negative	-0.13636
8	positive	0.35

Fig.14. example of data received from the sentiment analysis

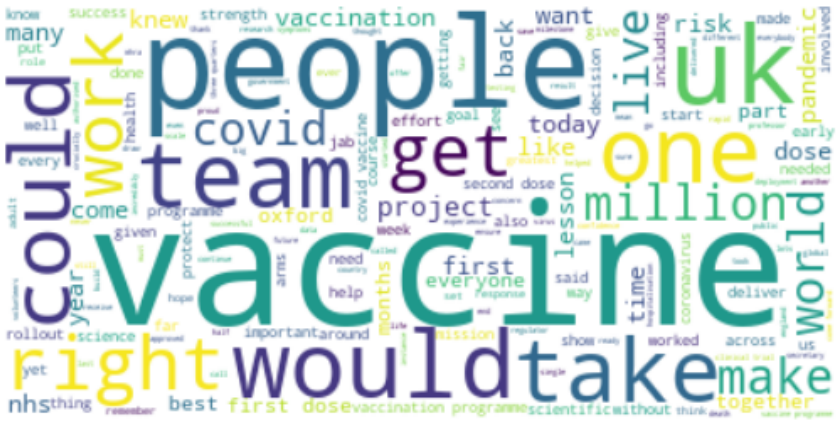

Fig.15. The results of the word cloud on 2-June-2021 for the paragraphs.

We can see in the word clouds that people are still talking about COVID-19 but less, related words to work are (work, projects, team) mentioned, the vaccination process, succes, also words that show closeness (people, everyone, close, together, team). The articles are more about the “let's do it” attitude (take, make, live, would, want, getting, strength, best).

Comparing both results, we can see how different the opinions are, people from Twitter are more realistic, still thinking about the aftermath of COVID-19, the vaccination, how the world is doing now (webinars, remote work and so on) meanwhile the articles from gov.uk are more refreshing, they intend to make people remember that right now there are better times, we can do more things, we are strong and much more. The articles give a more positive note than the tweets from people.

4. Experimental results

In this section, we will have more comparisons, we will use more days to compare and we will see how the results are. Different days will have different news, situations, that happen in the UK and it's the same for the people, they can have different things happening in their lives: there will be days where they got good news, so they will write a little more positive than the usual or maybe they will be neutral. If people get bad news, that's when they will write in a more negative way because they're thinking about it and their vision will also be noticeable in the tweets. We will see results where, even if we gathered tweets on COVID-19, there will be also mentioned other problems that happened in the UK (for example train accidents) or in people's lives (for example loss of a family member, accidents, problems at work).

For 5 June:

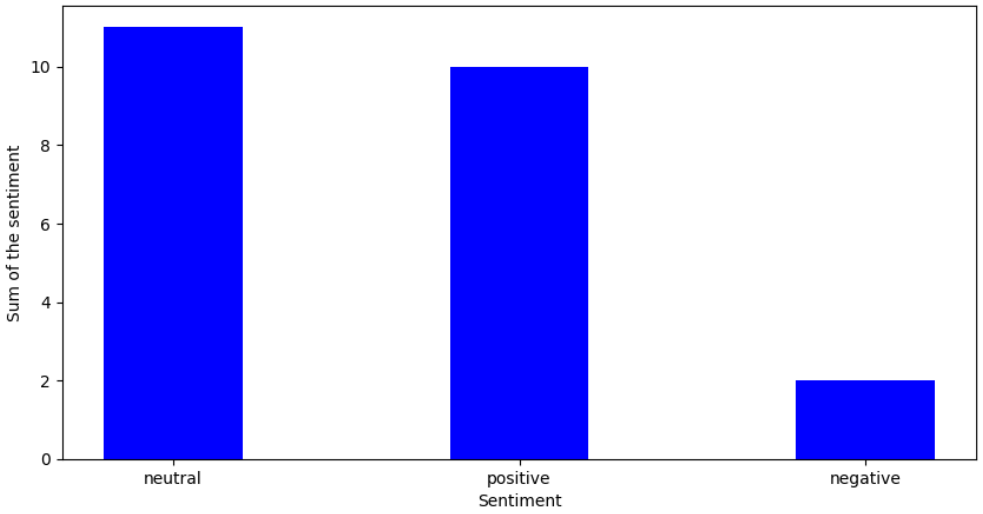

Fig.16. The bar plot resulted with the sentiment gathered from the tweets on 5 June 2021.

We can see how the tweets are written in a more positive manner, but also a neutral sentiment, the neutral one is predominating on this day. There aren't that many negative tweets but we can see that the sentiments don't vary that much compared with the other days. There are always many neutral and positive sentiments and much less negative sentiments.

0	neutral	0
1	positive	0.014899
2	neutral	0
3	negative	-0.15
4	neutral	0
5	neutral	0
6	positive	0.014899
7	positive	0.214286
8	neutral	0
9	neutral	0
10	positive	0.255556
11	neutral	0
12	positive	0.014899
13	neutral	0
14	neutral	0
15	positive	0.014899
16	positive	0.516667
17	positive	0.4375
18	positive	0.299091
19	neutral	0
20	positive	0.45
21	neutral	0

Fig.17. example of data received from the sentiment analysis

Fig.18. word cloud resulted with the sentiment gathered from the tweets on 5 June 2021.

We can see how people are still discussing everything related to COVID-19 (social distancing, deaths, covid19, pandemic, reopening, lockdown, vaccinated, reported) but at the same time they still get to talk about events (night club, reopening, restaurant, road) and some new things like train (an accident that happened in relation with a train).

Now for the articles from the same day:

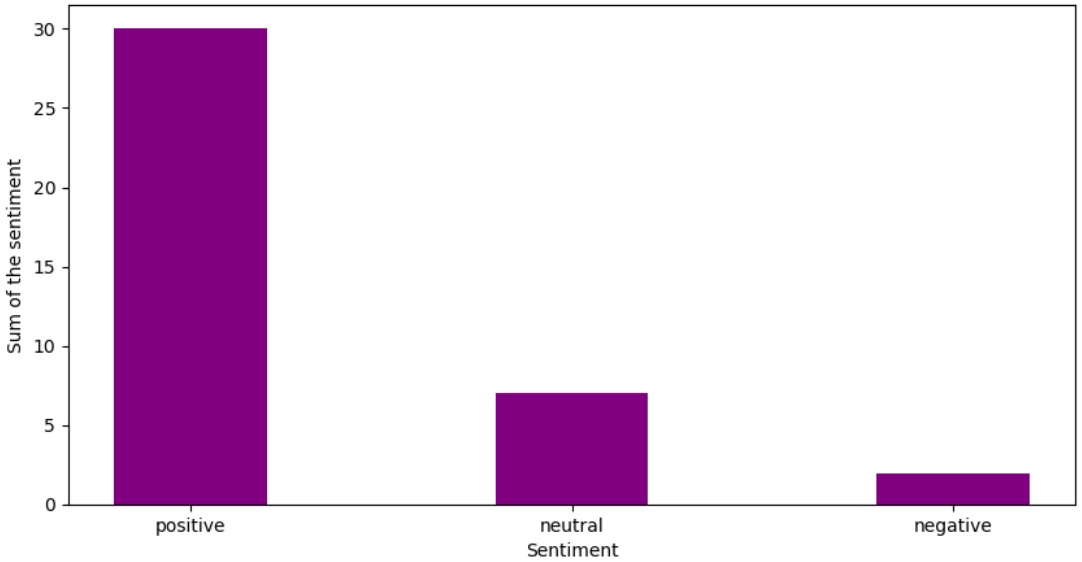

Fig.19. The bar plot resulted with the sentiment gathered from the news on 5 June 2021.

We can see how the articles are written in a more positive manner, even though there are some paragraphs with a negative connotation, there are not as many. The neutral sentiment is still present in some of the paragraphs but they're a little more present than the negative sentiments. Compared with the other days, the articles keep the same strategy, being really positive, avoiding being negative.

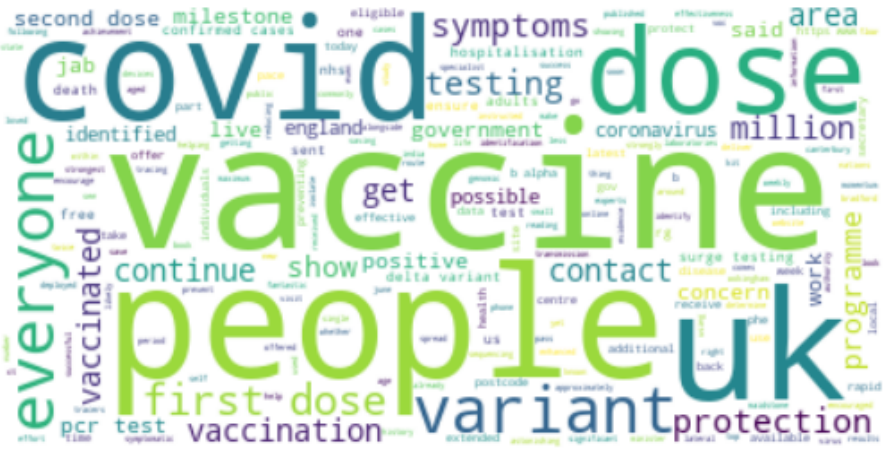

Fig.20. word cloud resulted with the sentiment gathered from the news on 5 June 2021.

We can see how the news is still writing about the vaccination, first and second dose, if they are vaccinated, about the variants, testing, symptomes, work, hospitalization, PCR test and so on.

We can see that both visions are predominantly positive but people tend to be more neutral than the news. People are still worried about COVID-19 and the aftermath with it but they're also starting to be more and more adapted to the current situation and for this reason, people tend to be more neutral than the news. The news want to still keep a calm view, they want to make people remember that the pandemic is going even better thanks to the vaccinations and thanks to the lockdowns and restrictions. The government wants to focus on solidarity, on the hobbies that people can start again, night activities but most important is the fact that we can change from “I” to “We”: we don't have to spend time alone and we can meet people/friends.

After collecting data from 5 consecutive days, we could make a box plot on the news and the tweets.

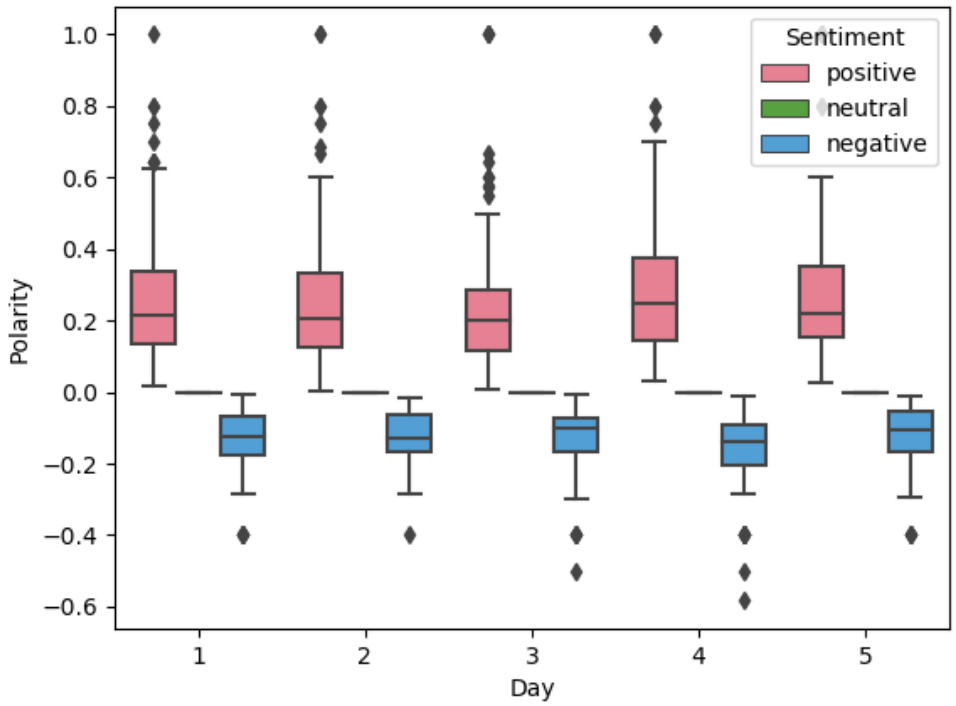

Fig.21. Box plot of 5 days - represents 1,2,3,4,5 June 2021 and are the sentiments from the articles of those days from the site.

In this box plot, we have the polarities, the sentiments of the articles gathered from a certain day. We can see that in the positive boxes, the articles have a median around 0.2-0.3, while the median for the negative boxes is around -0.1 close to 0, neutral. We can deduct from here that the majority of the negative paragraphs are not that negative, they try to not be when writing the news. Neutral sentiments don't show because all their values are 0. In all the days, the positive sentiment has grown and they almost reach a polarity of 0.7, close to the maximum. Those are named the upper and lower whiskers, they represent scores outside the middle. We can see that on days 1, 2 and 5, the positive boxes are positive skews, the other two are normal distributions. For negative sentiments, on day 1, 4 and 5 are normal distributions while the second day is a negative skew and on the third day is a positive skew.

An outlier is an observation that is numerically distant from the rest of the data, those outliers are the rombs from the plot.When reviewing a box plot, an outlier is defined as a data point that is located outside the whiskers of the box plot. We can see that the positive sentiment reaches the maximum on many days and there are many outliers on the positive site. On the negative part, there are also outliers, but not as many as on the positive side and they don't reach

the minimum, they only reach -0.6 and it's only one day that reaches that minimum.

Now the Twitter version:

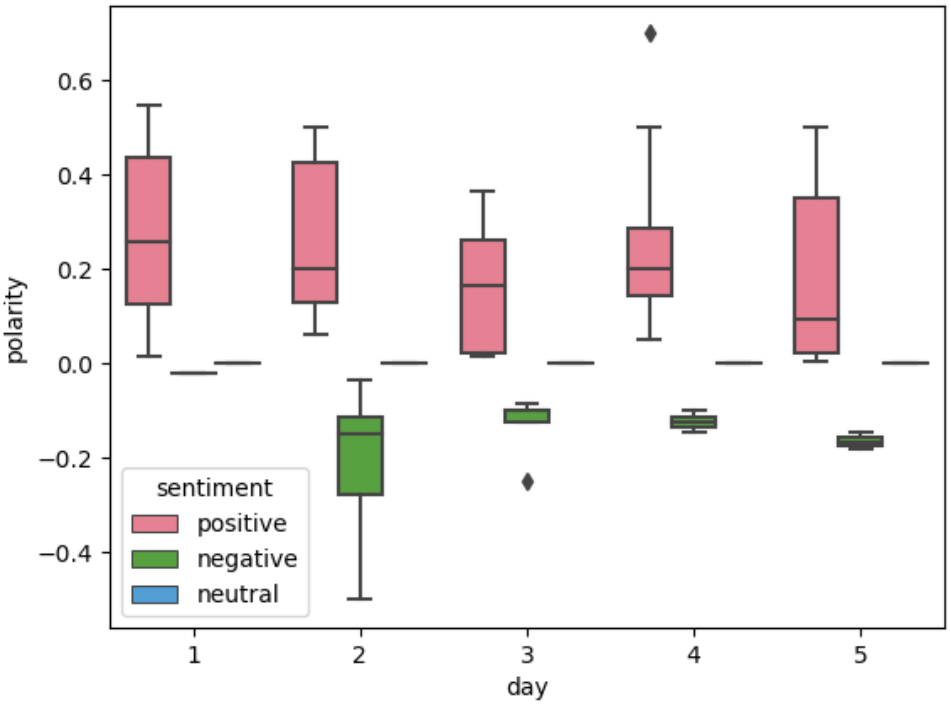

Fig.22. Box plot of 5 days - represents 1,2,3,4,5 June 2021 and are the sentiments from the tweets of those days from Twitter.

In this box plot, we can see how different it is compared to the box plot of the articles. The median for the positive side is around 0.1 to 0.3, a lot lower than the median from the sites, the median for the negative part, is around -0.1, not even reaching -0.2. We can see that on days 1, 2, 4 and 5, the positive boxes are positive skews, the 3rd one is a negative distribution, in this plot there are no normal distributions. For the negative side, there is only one positive skew, the one from the day 2, the ones from day 3,4 and 5 are normal distributions, at the same time, the one from the first june is almost nonexistent, it appears there was only one negative connotation on the first june. The outliers are almost nonexistent, they only appear on days 3 and 4, one in the negative side and the other one on the positive side.

From those two results, we can see how different their views are: articles tend to avoid being negative while the tweets from people are more realistic.

Now, one more example, 9 June 2021:

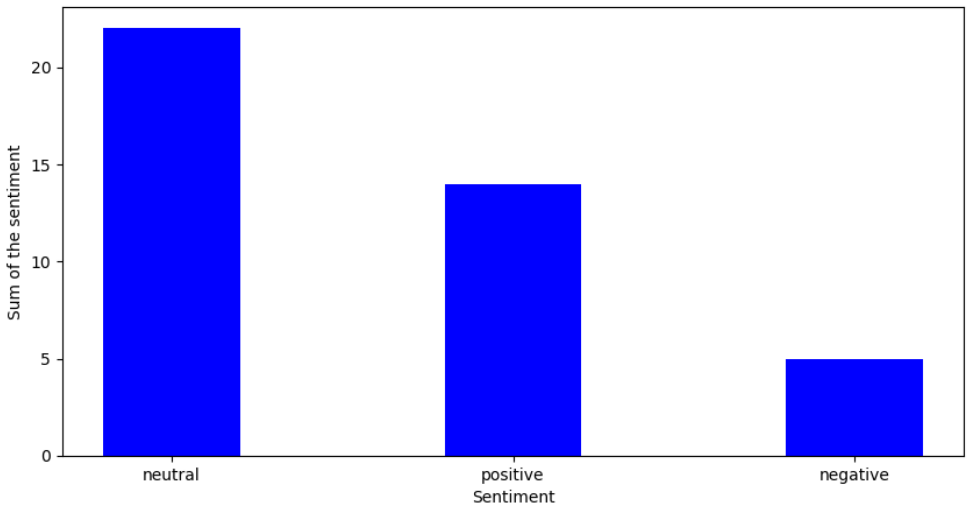

Fig.23. The bar plot resulted with the sentiment gathered from the tweets on 9 June 2021.

0	neutral	0
1	positive	0.5
2	neutral	0
3	neutral	0
4	positive	0.068182
5	neutral	0
6	neutral	0
7	positive	0.144444
8	neutral	0
9	negative	-0.45
10	neutral	0
11	neutral	0

12	neutral	0
13	positive	0.4
14	positive	0.4
15	positive	0.4
16	positive	0.4
17	negative	-0.28788
18	positive	0.5
19	neutral	0
20	neutral	0
21	neutral	0
22	neutral	0

Fig.24. example of data received from the sentiment analysis

We can see that the tweets tend to have a more neutral point of view, following a positive view and not as much negative. The same happens on the other experiments done.

Fig.25. word cloud resulted with the sentiment gathered from the tweets on 9 June 2021.

We can see how people are still discussing everything related to COVID-19 (pandemic, lockdown) but at the same time they still get to talk about their emotions (anxiety, fear) and some hobbies (books, webinars). The tone is more relaxed in some but at the same time they are more neutral. The negative feelings are reflected thanks to them sharing their emotions.

Now for the sites:

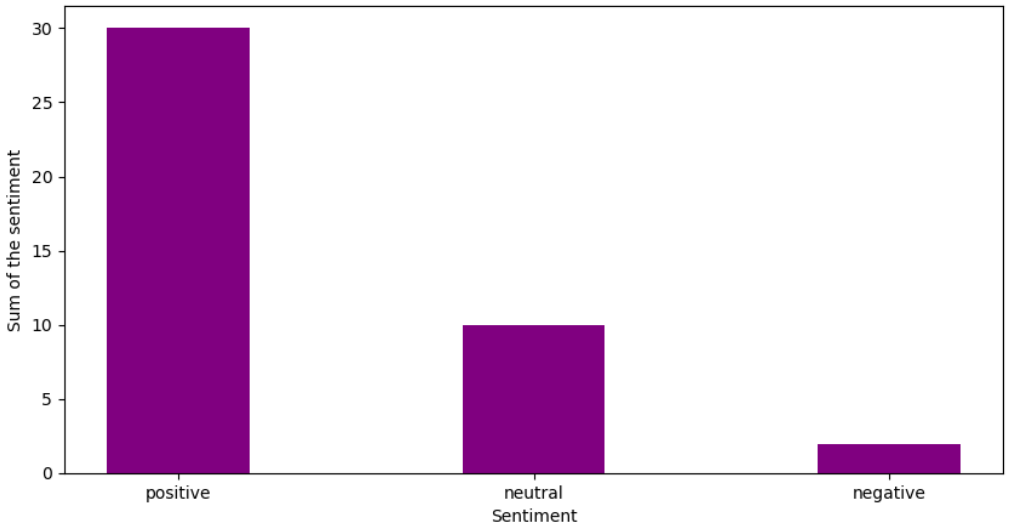

Fig.26. The bar plot resulted with the sentiment gathered from the news on 9 June 2021.

0	positive	0.068182
1	positive	0.190476
2	positive	0.15
3	positive	0.3
4	positive	0.2
5	positive	0.033333
6	positive	0.1
7	positive	0.5

8	positive	0.2
9	neutral	0
10	positive	0.3
11	positive	0.522222
12	positive	0.136616
13	positive	0.05
14	positive	0.1
15	neutral	0
16	positive	0.5
17	neutral	0
18	neutral	0
19	neutral	0
20	positive	0.084848
21	positive	0.136364
22	negative	-0.1

Fig.27. example of data received from the sentiment analysis

The articles are written in a more positive manner and follow the neutral sentiment, the negative feeling is almost nonexistent. Compared with the other days, the articles keep the same strategy, being really positive, avoiding being negative, it's almost identical with the graph of 5 June 2021.

Fig.28. word cloud resulted with the sentiment gathered from the news on 9 June 2021.

We can see how the news is still writing about the vaccination, symptoms, the variants, testing, health, PCR, pandemic, testing, test cases and much more. It still reminds people to protect themselves, follow the rules, to help people and so on.

After many experiments, in the end, we can see how similar the results are and the data are from different days. The articles are still written in a positive manner, avoiding to talk in a negative way. Meanwhile people are more neutral and positive, they still talk in a negative way when things aren't going well, they don't try to sugarcoat it like articles do.

Now we will do another box plot for another 5 days (6, 7, 8, 9, 10 june). First one will be the box plot of the articles of the site and the second one will be the box plot of the tweets of those days.

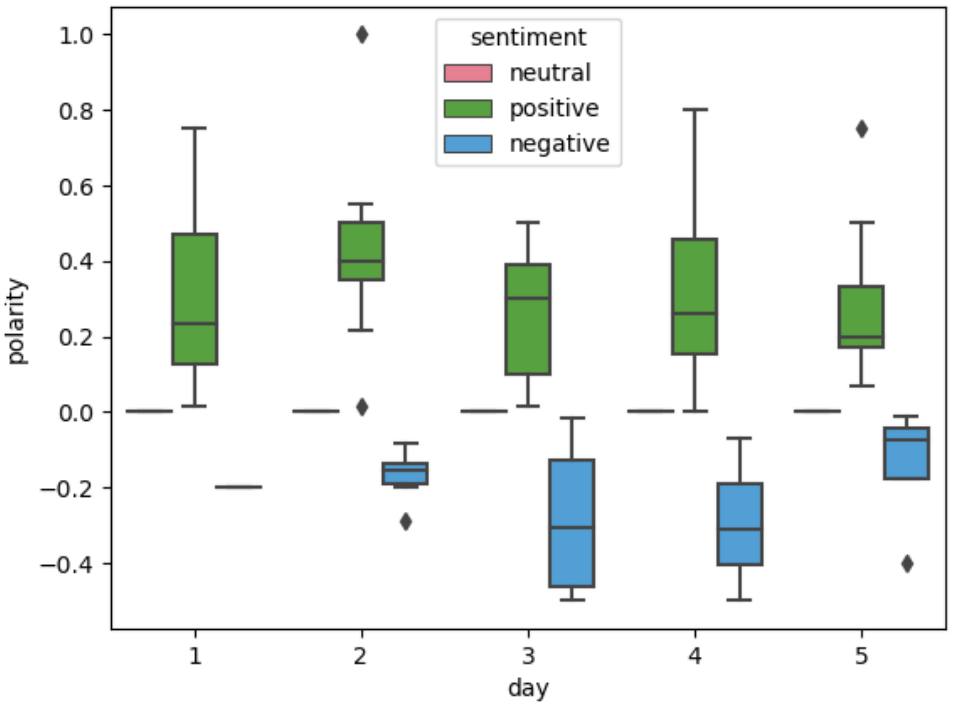

Fig.29. Box plot of 5 days - represents 6, 7, 8, 9, 10 June 2021 and are the sentiments from the articles of the sites.

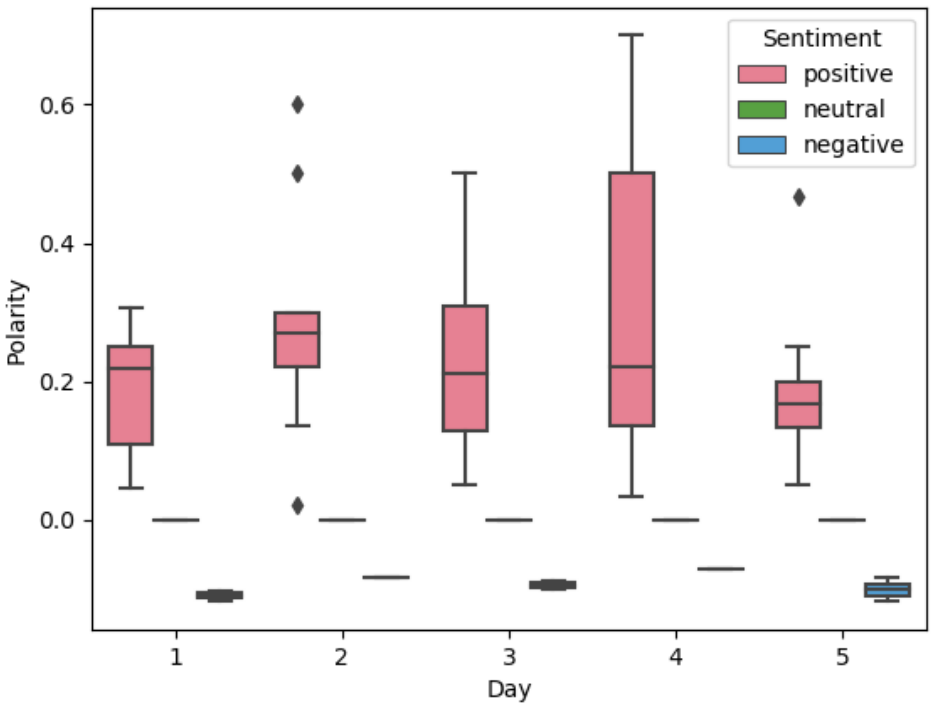

Fig.30. Box plot of 5 days - represents 6, 7, 8, 9, 10 June 2021 and are the sentiments from the tweets of those days from Twitter.

We can see that the same trend is still going: tweets are positive, there aren't many negative sentiments and the articles from the site are trying to send a really positive message while giving bad news with a really low polarity so they don't appear that bad. With the advancement of vaccination and people being immunized, people are slowly but surely starting to be more positive and less negative when it comes to the pandemic COVID-19.

5. Conclusion

Because of the constant decline in the number of COVID-19 cases and the lifting of the restrictions, the number of positive feedback greatly increased. Of course, people are still in shock, trying to adapt to normality. Some are completely burned out because of all that happens, some are still grieving the death of their family members and/or those of all the people around them who passed away due to the novel virus. But humanity is already taking big steps into returning to normal, trying to have fun while also being careful in closed spaces and trying to let loose after all these months of suffering. The statistics show the responses gathered in the last few days, but if we were to look through the ones appearing somewhere in the middle of 2020, the results would have been completely different, with a lot of people being confused, depressed, obsessed with every single opportunity they get to go out with friends. In other words, things escalated smoothly and in the right direction which shows how important the smallest things can be.

The Twitter API lets people extract just the last 7 tweets ending with the present day, which makes it difficult to keep on track or to back-track to the beginning of COVID-19. Twitter also has a Premium edition that lets you extract and analyze the tweets of over one month, more specifically 30 days, which is still not enough to make a statistics over the whole pandemic days.

With the results of the experiments we made, we can see that we reached the conclusion that people will write whatever they want on their Twitter and they will post their thoughts and feelings, those being negative, positive or neutral. Meanwhile the government of the UK keeps insisting (in those days) with vaccination and trying to send a message in a positive note, to remind people that in those days, everything is better and our lives will get better.

References

[1] Josep Sampé, Pedro García López, Marc Sánchez Artigas, Gil Vernik, Pol Roca-Llaberia, Aitor Arjona: Toward Multicloud Access Transparency in Serverless Computing. IEEE Softw. 38(1): 68-74 (2021).

[2]Serverless Computing –Architectural Considerations & Principles. January 2018 https://www2.deloitte.com/content/dam/Deloitte/tr/Documents/technology-media-telecommunications/Serverless%20Computing.pdf

[3]Hossein Shafiei, Ahmad Khnsari, Payam Mousavi, Serverless Computing: A Survey of Opportunities,Challenges, and Applications. June 2021 https://arxiv.org/pdf/1911.01296.pdf

[4]Announcing Amazon S3 - Simple Storage Service. Mar 13, 2006 https://aws.amazon.com/about-aws/whats-new/2006/03/13/announcing-amazon-s3---simple-storage-service/

[5]Serverless Computing, IBM Cloud Education, 20 April 2021 https://www.ibm.com/cloud/learn/serverless

[6]Cosmina Ivan, Radu Vasile, Vasile Dadarlat, Serverless Computing: An Investigation of Deployment Environments for Web APIs, 25 June 2019 https://www.google.ro/url?sa=t&rct=j&q=&esrc=s&source=web&cd=&ved=2ahUKEwj86d2bmf7wAhXtk4sKHY4SBF0QFjALegQIDBAD&url=https%3A%2F%2Fwww.mdpi.com%2F2073-431X%2F8%2F2%2F50%2Fpdf-vor&usg=AOvVaw3vmBWpi8AdErcw6SYajYjh

[7]Serverless Computing https://en.wikipedia.org/wiki/Serverless_computing

[8]Serverless Computing, British Columbia https://www2.gov.bc.ca/assets/gov/british-columbians-our-governments/services-policies-for-government/information-management-technology/information-security/information-security-awareness/thought_paper_serverlesscomputing_v_12.pdf

[9]David Wells, The rise of embarrassingly parallel serverless compute, 9 june 2020 https://davidwells.io/blog/rise-of-embarrassingly-parallel-serverless-compute

[10]James Beswick, Accelerating workloads using parallelism in AWS Step Functions, 20 March 2021, https://aws.amazon.com/blogs/compute/accelerating-workloads-using-parallelism-in-aws-step-functions/

[11]Josep Sampé, Lithops, a Multi-cloud Serverless Programming Framework, 9 March 2021,

https://itnext.io/lithops-a-multi-cloud-serverless-programming-framework-fd97f0d5e9e4
[12]Lithops, https://github.com/lithops-cloud/lithops
[13]Josep Sampé, Lithops Futures API Details, https://github.com/lithops-cloud/lithops/blob/master/docs/api_futures.md
[14]Twitter Sentiment Analysis using Python, 16 july 2020, https://www.geeksforgeeks.org/twitter-sentiment-analysis-using-python/
[15] Pang, Bo and Lillian Lee. 2008. Opinion mining and sentiment analysis. Foundations and Trends in Information Retrieval 2(1-2):1–135
[16]Introduction to Sentiment Analysis https://lct-master.org/files/MullenSentimentCourseSlides.pdf
[17]R. Feldman, "Techniques and applications for sentiment analysis," Communications of the ACM, vol. 56, no. 4, pp. 82-89,2013.
[18]Ahmad Anis, Getting Started with Sentiment Analysis using Python, https://cnvrg.io/sentiment-analysis-python/
[19] Maite Taboada, Sentiment Analysis: An Overview from Linguistics, 21 September 2015 https://www.annualreviews.org/doi/pdf/10.1146/annurev-linguistics-011415-040518
[20]Diana Kaemingk, Online reviews statistics to know in 2021, October 30, 2020 https://www.qualtrics.com/blog/online-review-stats/
[21]Kristen McCabe, 51 Customer Review Statistics to Make You Rethink Using Them, 28 September 2020, https://learn.g2.com/customer-reviews-statistics
[22]Bing Liu, Sentiment Analysis and Opinion Mining, 22 April 2012 https://www.cs.uic.edu/~liub/FBS/SentimentAnalysis-and-OpinionMining.pdf
[23]Christopher Potts, Sentiment analysis, May, 2019 https://web.stanford.edu/class/cs224u/2014/slides/cs224u-2014-lec15-sentiment.pdf
[24]Python Web Scraping, https://www.tutorialspoint.com/python_web_scraping/python_web_scraping_tutorial.pdf
[25]Kevin Sahin, Java Web Scraping Handbook, 26 July 2018 https://www.scrapingbee.com/download/webscrapinghandbook.pdf
[26]Osmar Castrillo-Fernández, Web Scraping: Applications and Tools, December 2015 https://data.europa.eu/sites/default/files/report/2015_web_scraping_applications_and_tools.pdf
[27]Emil Perssson, Evaluating tools and techniques for web scraping, December 2019,

http://www.diva-portal.org/smash/get/diva2:1415998/FULLTEXT01.pdf
[28]Daniel Glez-Peña, Anália Lourenco, Hugo López-Fernández, Miguel Reboiro-Jato, and Florentino Fdez-Riverola. Web scraping technologies in an api world. pages 788–796, 2013
[29]BeautifulSoup, https://www.crummy.com/software/BeautifulSoup/
[30]Ryan Mitchell, Web Scraping with Python, https://yanfei.site/docs/dpsa/references/PyWebScrapingBook.pdf
[31]Getting Started, About the Twitter API https://developer.twitter.com/en/docs/twitter-api/getting-started/about-twitter-api
[32]Getting Started, Getting Access to the Twitter API, https://developer.twitter.com/en/docs/twitter-api/getting-started/getting-access-to-the-twitter-api
[33]BeautifulSoup, https://pypi.org/project/beautifulsoup4/
[34]TextBlob, https://textblob.readthedocs.io/en/dev/quickstart.html