Covid-19 on the Web

Group 16

Weiqiang Guo
Student nr: 2748901
w.guo2@student.vu.nl

Lu Wang
Student nr: 2754874
l.wang2@student.vu.nl

Zheng Zhang
Student nr: 2748843
z.zhang10@student.vu.nl

ABSTRACT

This report gives a thorough explanation of the Covid-19 web project. The work is under the guidance of Peter Boncz. This project uses Common Crawl datasets to extract people's perspectives about Covid-19. This project is divided into two parts; the first looks into a global interest in Covid-19; the second concentrates on the UK and uses ranking, keyword extraction, and sentiment analysis approaches to investigate British people's opinions about Covid-19. The results are visualised by a static HTML web page (built by HTML/CSS/JavaScript). It allows users to get a sense of what British people care most when they talk about Covid-19, how they feel when discussing it, how much importance they place on it, and how they have changed over the last two years.

Keywords: Covid-19, Common Crawl, Sentiment analysis, Keywords extraction

1. INTRODUCTION

1.1 Motivation

From 31 December 2019, when the Wuhan Municipal Health Commission notified the pneumonia outbreak, to 26 September 2022, it has been exactly 1,000 days since the coronavirus outbreak entered the public consciousness. Over the past 1,000 days, the outbreak has had a profound impact on people's lives and economic activities in various countries. People have started to adapt to wearing masks for travel and gradually removed them, and we have become accustomed to regular nucleic acid testing. The length and depth of the impact of the coronavirus was unimaginable at the beginning of the outbreak.

With the aid of web crawl data provided by Common Crawl, we think it is worthwhile to review the dominant theories and trending topics and how they changed over time in each period during the preceding three years. These details are a representation of how British people, negotiate with, and are influenced by the corona virus. The findings of our work should provide users with an overview of how Covid-19's effects British people during the last three years.

1.2 Project Description

The objective of this project is to process and analyze three forms of crawl data from Common Crawl to extract websites from the UK that are linked to Covid-19 and to create a timeline where each time spot provides keywords and public sentiment in that month. The three types of data include,

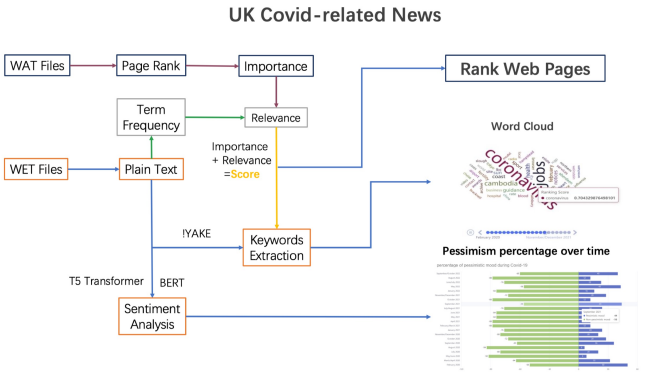

Figure 1: Project Outline

1. WARC: which stores the raw crawl data.
2. WAT: computed metadata for the data stored in WARC.
3. WET: extracted plain text content from the data stored in the WARC.

Due to the enormous amount(7090TiB in total) of data provided by Common Crawl and the limited computing power and financial resources we have, it is crucial to achieving two steps of data selection in order to complete this project. Since we are only interested in UK information, the first step is to extract web pages from the UK. The second step is to distinguish Covid-19 related pages and keep them. The amount of extraneous data can be significantly decreased through the two-step data selection, which allows us to have the data we can work with and commence our research.

For the analysis part, we have two parallel workflows which demonstrate in Figure 1. The first one uses filtered WAT data to perform Pagerank to rank the importance of the websites and use the topology map to show the citation networks of the websites. The second one uses text data extracted from filtered WET data, applying Spark NLP for sentiment analysis and then visualizing the outcome to demonstrate how British people think about Covid-19. Apart from that, we also combine the outcomes of the Pagerank and the term frequency to give more accurate results.

The structure of this report is as follows. The related research that illustrates the algorithms we will deploy in this project is described in Section 2. The research questions that we intend to address in this project are covered in Section 3. Extract-Transform-Load (ETL) pipeline in

Section 4.2 depicts the method of data processing. Section 4.1 is a general data investigation, and Section 4.2 is how we implement this project. The visualisation of result data is discussed in Section 5. The project's costs are shown in Section 6. A conclusion and discussion of the entire project are provided in Sections 7 and 8.

2. RELATED WORK

Data extraction, page rank, text summarising, sentiment analysis, keyword extraction, and visualization are the six components of this section.

2.1 Web Pages Importance

When we are using search engine, it is natural that the web pages get ranked so that people can get the most important and relevant web pages according to their queries [13]. In this section, we introduce how we measure the importance of a web page.

2.1.1 PageRank Algorithm

PageRank [13] is an algorithm that is commonly used by search engines to rank web pages. It is able to measure the importance of website pages by considering the number and quality of links to a page. In short, the more links to a site, and the better the quality of links, the more important the site is.

PageRank uses PR(PageRank) value to measure the importance of a website page. The higher the PR value is, the more important the corresponding website is. Assume that we have a web page A, its PR value can be calculated by the following equation,

$$PR(A) = \frac{(1-d)}{N} + d(\frac{PR(T_1)}{C(T_1)} + \cdots + \frac{PR(T_n)}{C(T_n)}) \quad (1)$$

where T_i is a page that links to page A, $C(T_i)$ is the number of outbound links of Page T_i, d is the damping factor that measures the probability that at any given moment, a user arrives at a page and continues to navigate backward, N is the total number of web pages.

2.1.2 GraphFrame

GraphFrame [10] is a package that provides DataFrame-based Graphs. It not only provides the functionality of GraphX [15], but also some new functionality based on Spark DataFrames. We uses this package to implement the PageRank algorithm.

2.2 Web Pages Relevance

2.2.1 Term Frequency

Apart from the importance of the web pages, we are also wondering about the relevance of the web pages to our keywords. For example, we are curious about how relevant each page is to the keyword Covid-19. Inspired by TF-IDF algorithm [14], we introduce the Term Frequency metric which is used to measure relevance.

The term frequency is intended to measure the relative frequency of a term within a document. It is defined by the following equation,

$$tf(t,d) = \frac{f_{t,d}}{\sum_{t' \in d} f_{t',d}} \quad (2)$$

where t represents the term, d represents the document, $f_{t,d}$ is the raw count of a term in a document, $\sum_{t' \in d} f_{t',d}$ is the total number of terms in document p.

The higher the tf value is, the more relevant the web page is to our keyword(covid).

2.3 The combination of the PageRank value and the tf value

In this project, we are playing as a search engine that is intended to find the most important and the most relevant web pages for our query(e.g. query is Covid-19). We first normalise the PageRank value and the tf value, which can eliminate the effects of dimensionality. Then, we use the weighted sum of the normalised PageRank value and the normalised tf value as the metric used to rank the web pages. We call this score as Ranking score of the web page. It can be calculated by the following equation,

$$PT(p) = normalised\_PR(p) + normalised\_tf(p) \quad (3)$$

where p is a web page.

With this ranking score, we can rank the web pages.

2.4 Sentiment Analysis

To gain a better understanding of the changes of human moods in a given period of Covid-19, various methods have been deployed to measure this. Attila Kiss [11] presented a model by analysing a Recurrent Neural Network, the model was built and taught using the libraries provided by tensorflow. Additionally, Khairiyah Mohamed Ridhwan used the sentiments returned from VADER lexicon-based classifier to find the sentiment changes throughout the Covid-19 period in Singapore, using the Python library 'SNSCRAPE'. John Snow Labs [3] also published a a pre-trained model named bert_sequence_classifier_emotion. This model is based on the Bidirectional Encoder Representations from Transformers model [12] with an extra linear layer and activation function. The model from John Snow was trained through Hugging Face and has a validation accuracy of 0.931 in the published documentation.

For this project, there are three demands in the sentiment analysis section: Need to process large amounts of data; Requires natural language processing; Requires a certain level of efficiency and can be deployed on DataBricks. For these demands, bert_sequence_classifier_emotion was selected as the main model of sentiment analysis.

In this part, we collect the relevant web pages in UK, filter them with Covid-19 related topic, pre-processing the whole plain text, summarizing the plain text into summaries using T5 Transformer, and perform an overall sentiment analysis using bert_sequence_classifier_emotion Model provided by Spark NLP. Then, we divided the output into two categories: pessimistic and non pessimistic. Finally, we can get the change of people's mood during Covid-19.

2.5 Keywords Extraction

It is essential to ensure that the extracted keywords correctly summarise each page's point of view in order to provide an accurate picture of the perspectives that have shaped the discussion surrounding the outbreak throughout time. There are various NLP methods available to extract keywords from filtered pages, and this project will use one of these methods, namely YAKE!. It has been shown to significantly outperform other unsupervised approaches for texts

of various length, languages, and domains when compared to 10 other state-of-the-art unsupervised methods and one supervised method.[8] The system has six main components: (1) Text pre-processing; (2) Feature extraction; (3) Individual terms score; (4) Candidate keywords list generation; (5) Data Deduplication; and (6) Ranking. For the data preprocessing, it is similar to most other natural language processing techniques, its early phase involves text cleaning, sentence splitting, text annotation, tokenization, and stopword detection. Second, feature extraction is devising a set of five features to capture the characteristics of each individual term. These are (1) Casing; (2) Word Positional; (3) Word Frequency; (4) Word Relatedness to Context; and (5) Word DifSentence. In the third step, the alogrithem heuristically combine all these features into a single measure such that each term is assigned a score S(w). This weight will feed the process of generating keywords which is to be taken in the fourth step. Here, it can be considered as a sliding window of 3-grams, thus generating a contiguous sequence of 1, 2 and 3-gram candidate keywords. Each candidate keyword will then be assigned a final S(kw), such that the smaller the score the more meaningful the keyword will be. Equation 1 formalizes this:

$$S(kw) = \frac{\prod_{\omega \in kw} S(w)}{TF(kw) * (1 + \sum_{w \in kw} S(w)))} \quad (4)$$

where S(kw) is the score of a candidate keyword, which is calculated by dividing the score S(w) of the candidate keyword's first phrase by the subsequent scores of the other words (in the numerator). The lower the S(kw) score, the more significant the keyword will be. [9]

3. RESEARCH QUESTION

- How to extract useful information from Common Crawl datasets?
- How to deal with so much data in the case that we have limited time and computing resources?
- How do we rank web pages in a reasonable way?
- How do we extract the opinions of people towards Covid-19?
- How British people's emotions changed over time?
- What British people are talking about Covid-19 in the past several years?

4. PROJECT SETUP

This section will describe the development and execution of the project. It starts with the input data from Common Crawl, and the initial investigation on the data. After this, the pipeline to process this data is created and described.

4.1 Input Data

All of the data that is used for this project comes from the Common Crawl dataset [1]. The Common Crawl builds and maintains an open repository of web crawl data since 2008, which can be accessed and analysed by anyone. The crawls is represented in three different data formats. The main structure of data stored in Common Crawl is shown in Figure 2.

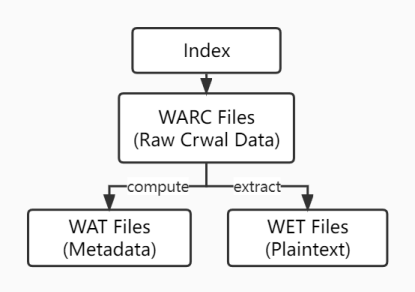

Figure 2: Data Format in Common Crawl

The Web ARChive (WARC) files store the raw crawl data. It contains the HTTP response, information about how that information was requested, and metadata on the crawl process itself. The WAT and WET formats are the summarisation of the WARC format.

The WAT format stores computed metadata for the data stored in the WARC, and is about a third the size of the WARC format. This metadata is computed for each of the three types of records (metadata, request, and response). If the information crawled is HTML, the computed metadata includes the HTTP headers returned and the links (including the type of link) listed on the page.

By contrast, the WET format stores extracted plain text content, and is about one-sixth the size of the WARC format.

There is another important file in the common crawl corpus named index, which stores the indexes to WARC files. With the help of the index, we can access WARC files, WAT files and WET files quickly.

In this project, we access the WAT and WET files rather than the WARC files. There are two reasons to explain why. First, the WAT and WET files contain all data we need, including the plain text content of the web pages, and links in the web pages. Second, WAT and WET files are much smaller than WARC, which means less resources will be consumed.

WARCIO library [7] provides a method to parse WARC files, WAT files and WET files. We access the WAT and WET with this tool. In addition, there are multiple information records in WAT and WET. We filter out specific WAT and WET information by using the URL as the main key.

4.2 Initial Data Investigation and Visualisation

The data archived in Common Crawl contains 63.94 billion web pages and 6790 TiB of uncompressed content crawled between February 2020 to October 2022. The data is composed by 22 data sets, with an average of 2.91 billion web pages and 308.63 TiB. Then, according to the top-level domain [5] of web pages, we classified the web pages according to the country. The average number of web pages in UK is 77 million per month, while 44 million for Netherlands, and 120 million for Germany.

In our project, we would like to know how the British people's perception of Covid-19 has changed over time. Hence, we pay close attention to the Covid-19 related web pages in

UK. The total number of web pages in UK which related to Covid-19 is 3,555,179, with an average number of 154,573. The detail is shown in Figure 3.

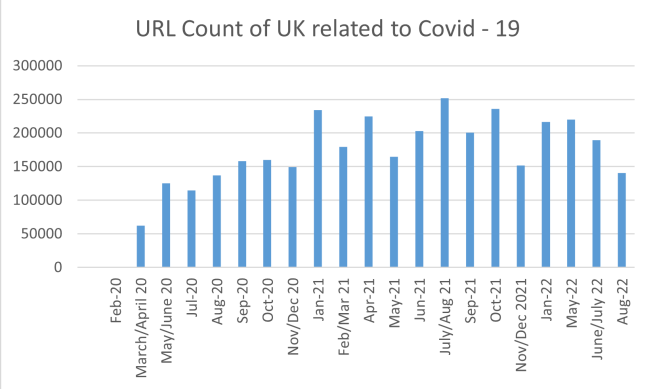

Figure 3: The number of URLs related to Covid-19 in UK

Figure 3 shows the trend exactly as we would expect. Little attention was paid to Covid-19 when it first appeared, but as Covid-19 grew in severity, attention to it increased and has remained at a high level since then.

Note that every data package(like February 2020 or May/-June 2020) has data of about 10-15 days. These days are either distributed in a separate month or two adjacent months. This is why some data packages are for one month and some packets are for two months. In addition, the data of some months are not included by Common Crawl, such as February to April 2022.

4.3 The Pipeline

This section discusses in more detail how to carry out the two-step data filtering and focuses on how we implemented the project and its pipeline.

4.3.1 Data Extration

In this part, there are three steps to extract the data. First, read the index file of WARC. Then, filter the index data in two steps to focus on the Covid-19-related web pages in the UK. Finally, change the index of WARC to WAT and WET. In this way, we can access the data in WAT and WET.

The Common Crawl data in this project was stored in Databricks File System(DBFS) [2]. The location path of index files is in the following format: "dbfs:/mnt/lsde/datasets/commoncrawl/cc-index/table/cc-main/warc/crawl=CC-MAIN-YYYY-WW/subset=warc/" where YYYY means the year, WW represents the serial number of the data package given by Common Crawl. The data packages selected for this project are shown in Table 1.

Table 2 shows some of the information contained in the index file. Then, we filter the index in two steps using url_host_tld and URL respectively. The url_host_host_tld contains top-level domains [5], which are those domains in the Domain Name System root zone. The Internet Assigned Numbers Authority(IANA) provides a list of Country code top-level domain(ccTLD). By querying the ccTLD, we can get the country corresponding to the top-level domain name. We focus on the United Kingdom in this project, so we filter the data with url_host_tld which equals to the UK. After limiting the scope of the UK, we need to focus on the content related to Covid-19 in the UK. A Uniform Resource Locator(URL) [6] is a specific type of Universal Resource Identifier, which follows the format: "https://domain.uk/news/title-covid-19-general". In this case, we perform a fuzzy match on covid and news to get the index of web pages related to Covid-19.

Finally, we change the index of WARC to access WAT and WET. Among files provided by Common Crawler, only WARC files have corresponding index files, while WAT and WET do not. However, since the sum of the WAT and WET volumes are smaller than half of the WARC, we choose to access WAT and WET instead of WARC. Therefore, we change the index of WARC to the corresponding WAT and WET. In addition, a WAT or WET file contains multiple pieces of information, so we matched their URL information with the corresponding WARC as the primary key to obtain accurate WAT and WET files. The python code of changing the index of WARC to WET and extracting the plain text is:

```
def process_wet(stream, url):
    for record in ArchiveIterator(stream):
        if record.get_header('WARC-URI') == url:
            wet_record = record.stream().read()
            return wet_record
            break
for index, row in pandas_df.iterrows():
    url = row[0]
    path = "/dbfs/mnt/lsde/datasets/commoncrawl/"
      + row[1].replace("/warc/", "/wet/")
      .replace("warc.gz", "warc.wet.gz")
    #read the whole plain text of WAT
    stream = open(path, 'rb')
    plain_text = process_wet(stream, url)
```

Table 1: Data Packages of Common Crawl

Package	Related Month	Package	Related Month
2020-10	February 2020	2020-16	March/April 2020
2020-24	May/June 2020	2020-29	July 2020
2020-34	August 2020	2020-40	September 2020
2020-45	October 2020	2020-50	Nov/Dec 2020
2021-04	January 2021	2021-10	Feb/March 2021
2021-17	April 2021	2021-21	May 2021
2021-25	June 2021	2021-31	July/August 2021
2021-39	September 2021	2021-43	October 2021
2021-49	Nov/Dec 2021	2022-05	January 2022
2022-21	May 2022	2022-27	June/July 2022
2022-33	August 2022	2022-40	Sep/Oct 2022

Table 2: Information in the Index file

Attribute	Meaning
url	The url of the web page.
url_host_name	The domain of the web page.
url_host_tld	The top-level domain, related with country.
fetch_time	Timestamp.

Unlike WET fileswhich contain plain text content directly, WAT files store metadata and some header information. The metadata is written in the form of JavaScript Object Notation(JSON) format [4]. Therefore, we need to read the JSON using the python JSON library. Alter parsing the JSON data, we can extract links of web pages.

4.3.2 *Ranking the web pages*

After filtering the data, we rank the web pages. First, we apply PageRank algorithm to compute a PageRank value for each web page which can measure the importance of a web page. However, due to the number of web pages we get is not very big (10000 web pages a month), there is no edge between web pages. As a result, every page get the same PageRank value which means that each web page is treated equally important.

Second, we compute Term Frequency on the plain text content extracted from WET files for each web page. Term Frequency is the frequency of Covid-related terms ocurring on a web page. In our case, we take covid and coronavirus as the covid-related terms. Then, we combine PageRank value and Term Frequency to get a ranking score to rank the web pages.

4.3.3 *Keywords Extraction*

After ranking the web pages, we can do keywords extraction on the plain text content and combine it with the ranking score we get before to generate a word cloud for each month to see what were British people talking about Covid-19, and how they changed over time. In the word cloud, the size of words is determined by the number of their occurrences multiplied by the value of the corresponding ranking score.

The reason why we want combine the ranking score with keywords is that we think the more important a website is, the more important the keyword extracted from it.

In this part, we first extract plain text content from WET files for each web page, and then apply !YAKE model to generation a keyword for a web page. Furthermore, we generation a word cloud for each month based on the keywords we get.

4.3.4 *Sentiment analysis*

Apart from keywords extraction, we can also do sentiment analysis on the plaintext. We first summarize plain text content of each web page using T5 Transformer model then apply sentiment analysis with bert_sequence_classifier_emotion model based on the summarization. As a result, we get the distribution of sentiment about Covid-19 of the British people for each month, and the percentage of pessimistic and optimistic. With this sentiment analysis, we can get an overview of British people's attitudes towards Covid-19 for each month, and see how their opinions changed over time.

5. VISUALISATION

The visualisation of our data product is shown in a static website, using HTML, CSS and JavaScript. The structure of the visualisation is as follows. The URL of website is: https://8mile313.github.io/LSDE2022-G16/.

5.1 Homepage

This section is an introduction to our project, a description of each data product, and information about the data selected for the project. In the top right corner of it, there are links to jump to each data products, such as Public attention, keywords and sentiment analysis. It is shown in Figure 4.

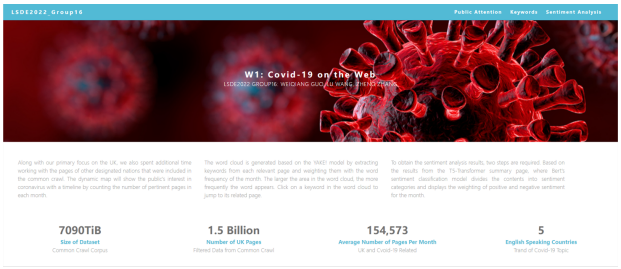

Figure 4: Homepage

5.2 Word Cloud

The visualisation also include a Word Cloud for each month. This illustrates what British people were discussing about Covid-19. The Word Cloud as seen in Figure 5 contains the words of the month which are extracted from each relevant pages. The size of the word in the chart is determined by the number of their occurrences multiplied by the value of the ranking score. Note that, in our case, every page has the same PageRank value, which means that they are treated equally important. Hence, the ranking score of a web page is determined by the Term Frequency.

There is a timeline controller in the bottom of the word cloud page, which allows the user to see different month's word cloud. The word cloud is set as auto play as default, which means that it will automatically display the word cloud for the different months. The user can pause the word cloud by clicking on the pause button on the left.

Apart from timeline, the word cloud has another characteristic. If a user clicks on a word of interest, the web page will jump to the corresponding page that generates the word. With this feature, users can not only see the high frequency words, but also get to know the corresponding web pages for more detailed information.

Figure 5: Word Cloud

With word cloud pages, users can get an overview of what were British people talking about Covid-19, and how they changed over time.

5.3 Sentiment Distribution

The sentiment distribution of United Kingdom relevant to Covid-19 is shown in the Figure 6. The sentiment target is derived from summaries of each relevant page of each month from February 2020 to October 2022. The percentage of Non-pessimistic mood is shown by the green bar on the left, while the converse is on its right. By observing the bar graph, we can observe the mood of Britain towards the COVID-19.

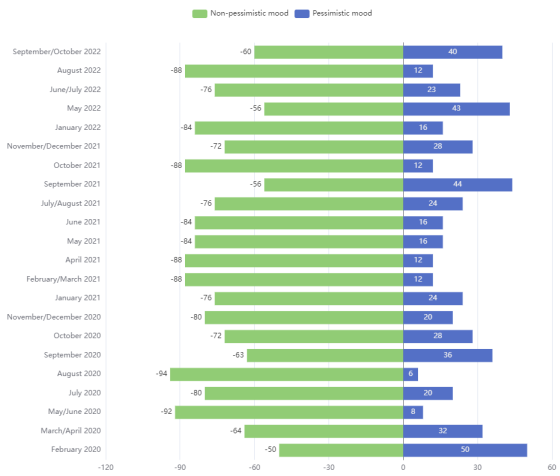

Figure 6: Distribution of Sentiment

5.4 Public Attentions

Apart from British people's opinions towards Covid-19, it is also interesting to see the public attentions on Covid-19 around the world, and how they changed over time. Hence, we select several representative countries, such as United States, United Kingdom, Australia, China, Brazil, Netherlands, etc, to find out how much public attention is paid to Covid-19 around the world and how they have changed over time. We shows the changing trend of attentions by the world map, as shown in Figure 7.

Figure 7 shows the number of Covid-related web pages which represents the attention of people on Covid-19 each month. For example, In June 2021, United Kingdom has 351907 web pages related to Covid-19. In this map, we use different colours to represent different levels of attention. The warmer the colour, the higher the attention, vice versa, as shown in the bottom right corner.

The timeline controller at the bottom of the map enables users to view a map from a different month. The default setting for auto play means that the map will automatically change to reflect the current month. Users can then observe how Covid-19 has brought attention over time in the nations they have selected. By using the pause button on the left, users can also pause the map switching.

6. CALCULATION OF PROJECT COST

At the beginning of our project, we estimated our cost. It is necessary to make an estimation due to the fact that we have limited budget. Now we have finished our project, we can give a more accurate estimation of our project cost.

We first look at how big our data is and how much computing time we need. We need to access WAT and WET files related to the Covid-19 subject in the United Kingdom. Their size volume is about 41 GB(27.3 GB for WAT files, 13.7GB for WET files), and there are about $3,519,340$ web pages. We need to use them to measure the importance of each web page and do sentiment analysis. For WAT files, we need to extract links on every web page, build a graph whose nodes are web pages, and compute PageRank value for every node. After extracting links, we probably get a much smaller dataset. Let us assume its size is one-tenth of the original dataset, which is about $27.3 * 1/10 = 2.73$GB. For WET files, we only need plain text information which accounts for roughly 90% of the data, that is $13.7 * 0.9 = 12.33$GB. In addition, we need to store the final results of our pipeline, such as keywords, people's emotions, etc. It is about 3GB. In a word, we have about $12.33 + 2.73 + 41 + 3 = 59.06$GB data needed to be stored in S3 buckets.

6.1 EC2 Cost

The price for an i3.xlarge server is $0.31 per hour. Our timeline and planning shows that we run about 315.5 hours, which would cost about $195.61. Every i3.xlarge instances have 30.5 GB memory and two CPU cores.

6.2 S3 Cost

For S3 Standard plan, the price for the first 50TB stored for one month is $0.0245 per GB. The price for PUT, COPY, POST, LIST requests (per 1,000 requests) is $0.0054. The price for GET, SELECT, and all other requests (per 1,000 requests) is $0.00043.

In our case, we currently need to store about 59.06 GB of data which would take about $\$59.06 * \$0.0245 = \$1.447$. In addition, we take the cost of requests into account. Roughly, we need to access all data stored in S3 buckets which means that we need to perform LIST and SELECT at least once for each WAT file and WET file. Hence, the price paid for requesting is about $(\$0.0054 + \$0.00043)/1000 * 3519340 \approx \$20.52$. The total of S3 cost is $\$20.52 + \$1.447 \approx \$21.97$.

6.3 Project Cost in Total

Summarising the calculations above, we get a very rough estimation which is about $\$195.61 + \$21.97 = \$217.58$.

7. CONCLUSIONS

To sum up, our work gives an overview of people's attention to Covid-19 and how it changed over time. We have built an ETL pipeline to filter the data from Common Crawl datasets, which is a challenge in our project since the dataset have a huge amount of data. Next, we extract useful information from filtered data. We investigate people's attention on Covid-19 around the world. In addition, we extract keywords and emotions from United Kingdom web pages. More importantly, these things are repeated each month. With this method, we can see only people's opinions about Covid-19, but also how they changed over time. Finally, the results are visualised in a static HTML website. The public attention of each month is shown on a world map, as you can see from Figure 7. The keywords are shown in a form of word cloud, as you can from Figure 5. Through the

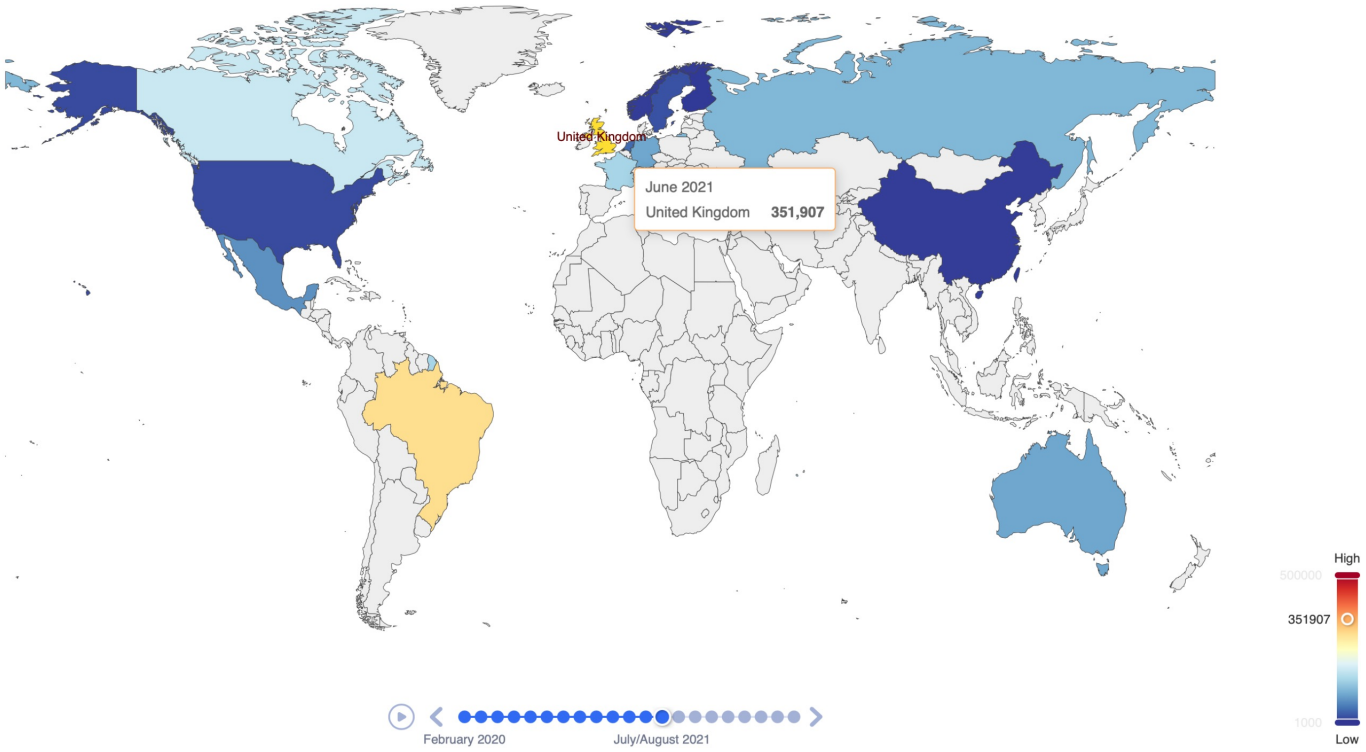

Figure 7: Public Attention Map

visualization, one can see very visually how people perceive Covid-19 and how these perceptions have changed over time.

8. DISCUSSION AND FUTURE WORK

Although we have good results based on the work we mentioned above, there is plenty of room for improvement.

First, in our case, the number of web pages each month is not vast enough, which causes every web page to have the same PageRank value. If we get more time and more resources, we could apply the PageRank algorithm to bigger datasets. It might be possible that we get a more connected graph, which means different web pages may have different importance(PageRank value).

Second, we could analyze more countries. Currently, we select several representative countries to show the trend of the world. It is inevitable to miss other countries' information. If we analyze more countries, we could get a more comprehensive picture.

Third, in our project, we filter the data by scanning the URLs. The data we get is inevitably skewed. It turns out that the data we get is about 1% of all Covid-related web pages. If we get more time and resources, we could filter the data by scanning the entire text content of each web page. In this way, our analysis will become more accurate since we get more data.

Fourth, we could spend more time on the visualizations, and then give users a more intuitive and convenient visualization.

In a word, there are a lot of improvements we can do, however, due to the fact that we only have limited time and resources, we can only take this work as the future work of our project.

9. COOPERATION

Our team did well in terms of communication and collaboration. We struggled with this project and learned a lot from it. The specific work of each person is shown in the following Table 3.

Table 3: Group cooperation

Task	Who
Initial data investigation	Weiqiang(mainly), Lu
Project plan	All
Data analysis	Lu(mainly), all
WAT extraction	Weiqiang
WET processing	Lu
Keyword extraction	Zheng
Ranking web pages	Weiqiang
Sentiment analysis	Lu
Paralleling the code	Weiqiang
Visualisation	Zheng(mainly), all

10. PROJECT SCHEDULE

The schedule for our project is depicted in the Figure 8 below, where data processing and analysis took up the majority of our time.

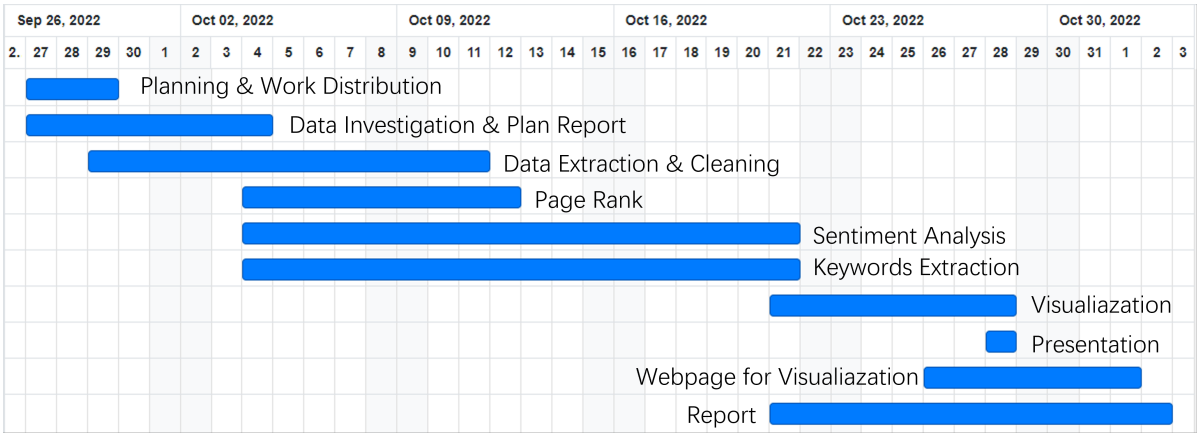

Figure 8: Project Schedule

11. REFERENCES

[1] Common crawl: Build and maintain an open repository of web crawl data. https://commoncrawl.org/.

[2] Dbfs: Databricks file system. https://learn.microsoft.com/en-us/azure/databricks/dbfs/.

[3] John snow labs: Nlp and ai in health care. https://www.johnsnowlabs.com/.

[4] json library: Json encoder and decoder. https://docs.python.org/3/library/json.html.

[5] tld: List of internet top-level domains. https://en.wikipedia.org/wiki/List_of_Internet_top-level_domains.

[6] url: The components of a url. https://www.ibm.com/docs/en/cics-ts/5.1?topic=concepts-components-url.

[7] Warcio: Warc (and arc) streaming library. https://github.com/webrecorder/warcio.

[8] R. Campos, V. Mangaravite, A. Pasquali, A. Jorge, C. Nunes, and A. Jatowt. Yake! keyword extraction from single documents using multiple local features. *Information Sciences*, 509:257–289, 2020.

[9] R. Campos, V. Mangaravite, A. Pasquali, A. M. Jorge, C. Nunes, and A. Jatowt. Yake! collection-independent automatic keyword extractor. In G. Pasi, B. Piwowarski, L. Azzopardi, and A. Hanbury, editors, *Advances in Information Retrieval*, pages 806–810, Cham, 2018. Springer International Publishing.

[10] A. Dave, A. Jindal, L. E. Li, R. Xin, J. Gonzalez, and M. Zaharia. Graphframes: an integrated api for mixing graph and relational queries. In *Proceedings of the fourth international workshop on graph data management experiences and systems*, pages 1–8, 2016.

[11] L. Nemes and A. Kiss. Social media sentiment analysis based on covid-19. *Journal of Information and Telecommunication*, 5(1):1–15, 2021.

[12] D. Nozza, F. Bianchi, and D. Hovy. What the [mask]? making sense of language-specific bert models. 03 2020.

[13] L. Page, S. Brin, R. Motwani, and T. Winograd. The pagerank citation ranking: Bringing order to the web. Technical Report 1999-66, Stanford InfoLab, November 1999. Previous number = SIDL-WP-1999-0120.

[14] A. Rajaraman and J. D. Ullman. *Mining of massive datasets*. Cambridge University Press, 2011.

[15] R. S. Xin, J. E. Gonzalez, M. J. Franklin, and I. Stoica. Graphx: A resilient distributed graph system on spark. In *First international workshop on graph data management experiences and systems*, pages 1–6, 2013.