***Web Science Coursework Report***

*Alana Grant 22390384, heather cook 2375346*

[*https://github.com/hdunlop310/Web-Sci-2021*](https://github.com/hdunlop310/Web-Sci-2021)

2021

**Section 1: Introduction**

**a. Describe the software developed with appropriate details; if you have used code from elsewhere please specify it**

Software used – PyCharm, MongoDB    
Code used – Josemon's starting guide code to partly process tweets, Natural Learning Toolkit for removing stopwords, Sklearn for help clustering tweets.

**b. Specify the time and duration of data collected.**

We ran code.py for 1 hour.

**Section 2: Data crawl**

**a. Use Twitter Streaming API for collecting 1% data**   
**i. Specify the APIs used**

**1. Please do not include entire code here; just main description of the function.**

We used the Tweepy streaming API that was given to us from the starting code guide provided, the code accessed twitter through the authentication keys and began to collect tweets. The tweets were gathered based on a list of keywords we had given, the list was made up of different words based on UK politics and the COVID impact on the UK. As the tweets were gathered by the API, each one was processed through a function called process\_tweets which helped to clean the tweet (e.g. get rid of emojis, etc) and pick out the relevant information fields. Afterwards, the tweet was inserted into a collection called ‘March21st’ in MongoDB in order to help us extract data from the collection of tweets.

We created a new python file called partOne which helped count the number of tweets, retweets, images, etc collected by accessing the database and iterating over the different data fields.

**2. Along with a short description/justification**

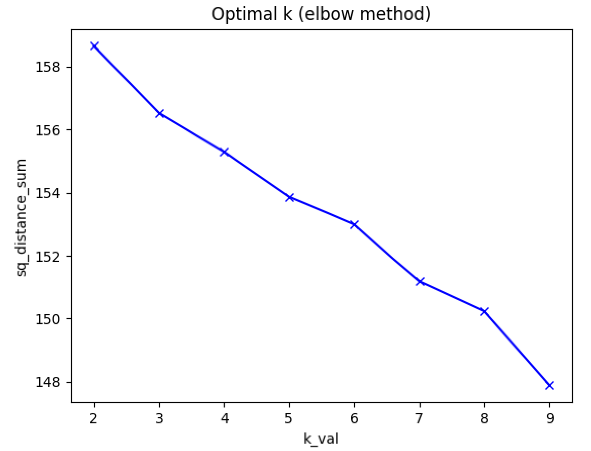
We can justify using this method seeing as the sample code was already provided so it would make most sense to make use of it and just tweak it to our requirements. The tweets we have gathered collect the appropriate data seeing as it looks for UK tweets as requested, the keywords searched for are topical and current for this climate therefore can generate a lot of data. The hashtags searched for are also topical for example we have #COVID, #vaccine and #BorisHasFailedTheNation which are hashtags that have been popular recently so will also aid with the Twitter scraping.

**Describe the seed crawl data used – (some given in the sample code) Users, hashtags, words, location etc.**

Data collected:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Total** | **Streaming API** | **No of retweets** | **No of quotes** | **No of images** | **No of videos** | **No of verified** | **No of geo-tagged** | **No of place/location objects** |
| 179,991 | ?? | 42773 | 207 | 41208 | 115 | 2039 | 57694 | 111990 |

**b. Specify data grouping methods and associated statistics Similarity measure used - briefly explain the method.**

For the second part we decided to combine some of the sample code with our own code in order to have only geo-enabled tweets, ensuring it was just UK tweets that our stream was gathering.

Our method for grouping similar data was through K-means, which is an algorithm that attempts to group similar data together in the form of clusters, the K number represents the number of groups.

The first step was to represent each tweet as a numerical vector, we used the Sklearn’s TfidfVectorizer function to easily compute the calculations. Once this was done we were able to plot each vector representation on a graph in order to determine the appropriate number of clusters this was done using the Elbow method.

From the graph there was a bit of a decrease/dent around 7, so we attempted to group the data into 7 clusters. Once the clusters had been created, we were able to put the tweets into a dictionary and insert them into individual collections within the database.

Data collected:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Total** | **Formed groups** | **Min size** | **Max size** | **Average size** |
|  |  |  |  |  |

**Discuss the effectiveness of similarity, grouping strategy, nature of Twitter data etc.**

Due to the nature of tweets, they contain a lot of redundant words that are not needed when trying to analysis data. In order to reduce the effects of these words when clustering. We implemented the NLTK which has a list of common English stop words, by doing this the clusters became clearer and more accurate.

Due to only gathering Geo enabled tweets, this dramatically reduced the number of live tweets we gathered, because of this we increased the number of keywords and hashtags being searched for. The coursework requested that it be only UK tweets gathered so this was the most effficent way of determining each tweet was from somewhere in the UK.

K-means as a measure of similarity has its advantages, it’s very simple to implement, its scalable and fast for large datasets which came in handy for our data and it’s visual representation of the K-mean through the Elbow method made it easier to visualise the dataset at hand. However, there are some down sides to using K-means, it can be sensitive to outliers and manually choosing the K-means can be difficult and lead to human error. In our case the K-means groups would range from 4 – 64, which means the data is more spread out and we should reduce the number of clusters used to try get a more even amount in each cluster.

Grouping the tweets into different collections, was the most logical approach from the given K-means clusters. It was easy to pull the required data fields and place them into individual collections. This allowed us to see each of the cluster’s data easily and identify the different grouped topics in each one. In each collection the data was stored as a dictionary, this was so the data would be well structured and the information would be clear to get to. Due to the dictionary key being the name of the different fields of a tweet, it made analysing the collections even quicker and this also helps for easy look up of data.

**c. Enhance the crawling using the hybrid architecture of Twitter Streaming & REST APIs**

In order to enhance the crawling we identified from section b the groups which were most identifiable from their cluster, the tweets that had clearer groups were those about COVID. This meant that we could prioritise the group of COVID tweets over the rest that we had gathered and priorities any of our keywords that were about the pandemic over those about general politics. One way we did this to enhance the crawler and remove some redundant data was to reduce the number of keywords we searched for.

Another way that we enhanced the crawler was by gathering tweets from specific users that had been posted within the last day. By collecting tweets from the UK health secretary, the first minister and other UK science stat accounts we were targeting our crawler to gather tweets about the pandemic and were located in the UK.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Total | Streaming API | REST API | Redundant tweets | No of quotes | No of retweets | No of geo-tagged | No of images/vids |
| 205 | 92 | 74 | 0 | 20 | 54 | 92 | 56 |

**Discuss the results – effectiveness of the approach**

This approach proved to be effective for gathering more specific data clusters, this making the data easier to analysis as it was just from one topic in the UK instead of a more general range. Efficiency wise, this did reduce the number of tweets that the crawler gathered but by doing this the clustering groups were more accurate and better reflected the different groups of data.

By having our previous crawler and our enhanced crawler run at the same time for the same hour, we were able to make a direct comparison about the efficiency of the enhanced crawler from the different data gathered. We felt Sunday afternoon would be a good time to run the crawler since majority of people are off work and Twitter is generally more activate over the weekend period than during the week work hours so we could gather a good amount of tweets for analysing.

One issue found from the previous crawler was a lot of redundant tweets that didn’t make sense due to some stop words not being removed. To overcome this issue we removed more stop words and punctuation to make our clustering more efficient.

When we did cluster the data thanks to the removed noise it allowed the groups to become clearer for us to analysis increasing efficiency. We also felt the need to reduce the number of clusters used to group the data so we only did 5 groups, this made the data more evenly spread out unlike the previous crawler.

Graphing geographical information

Master doc of all cities in the UK: <https://simplemaps.com/data/world-cities>. (have to give credit)

Other list <https://www.townslist.co.uk/>

March21New (from code.py)

Cluster0

quote tweets = 2508

retweets = 0

geotagged = 40231

location objects = 78956

verified = 1820

videos = 88

images = 33203

Cluster1

quote tweets = 93

retweets = 0

geotagged = 2555

location objects = 5587

verified = 46

videos = 0

images = 863

Cluster2

quote tweets = 34

retweets = 0

geotagged = 13025

location objects = 24521

verified = 159

videos = 27

images = 7141

Cluster3

quote tweets = 0

retweets = 0

geotagged = 796

location objects = 1103

verified = 2

videos = 0

images = 0

Cluster4

quote tweets = 0

retweets = 0

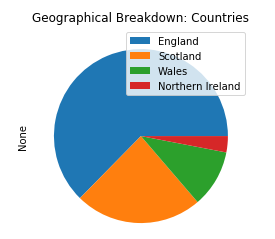
geotagged = 1087

location objects = 1823

verified = 12

videos = 0

images = 1



**Section 4: Event detection (group project)**

**a. Identify important events in the clusters**

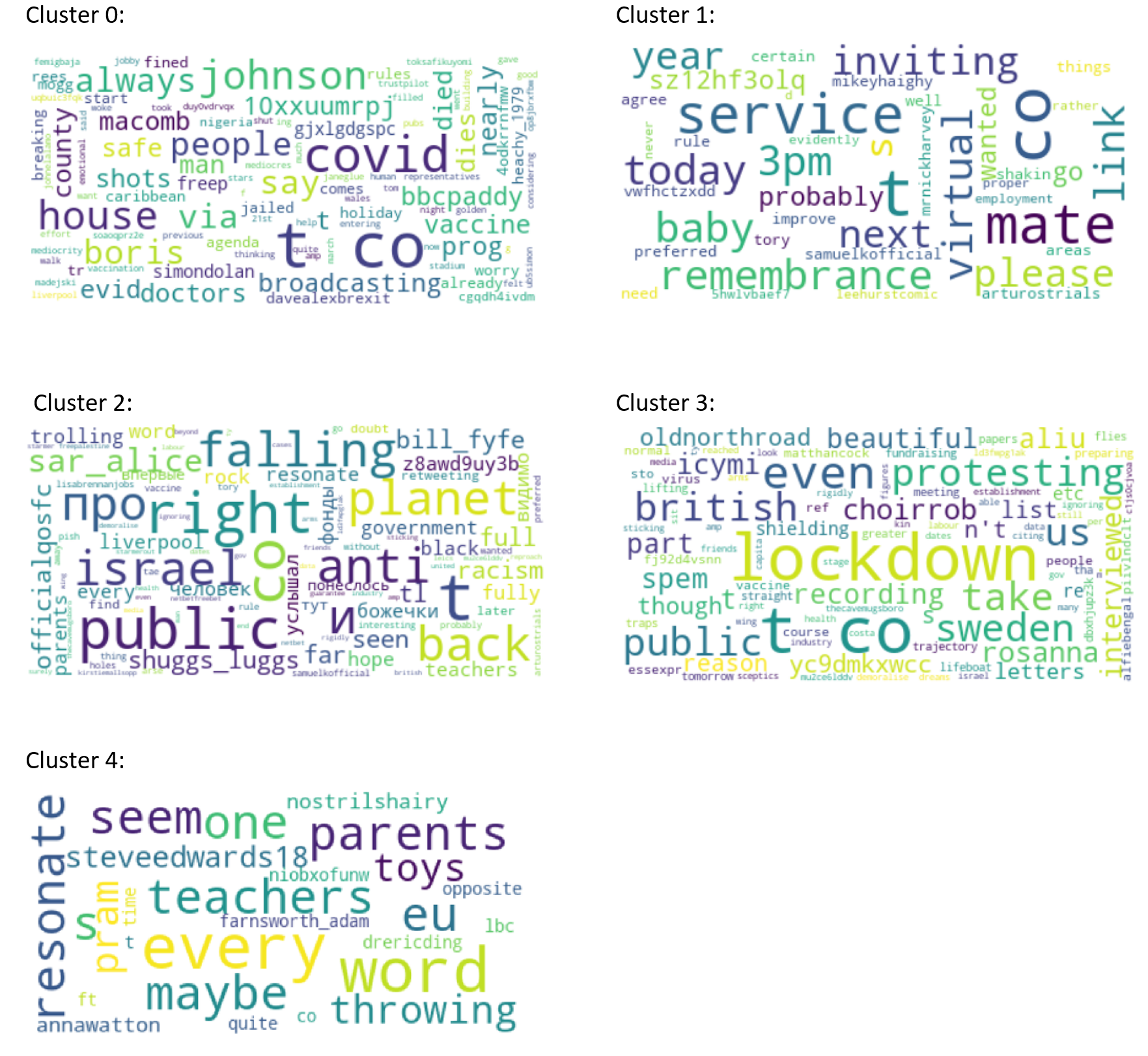
These events are from the data gathered from the enhancedCrawler collection. In order to identify events we clustered the data into 5 groups and since there wasn’t a large amount of data gathered due to the enhancedCrawler we could just manually detect the events.

We used a wordcloud to see which clusters grouped what tweets, some clusters are more obvious than others for example cluster 1’s event is a virtual remembrance service on today at 3pm.

Cluster 0 is a bit more vague but due to the words ‘boris’, ‘johnson’, ‘vaccine’ and ‘covid’, we can assume the event is about the prime minster and recent news about how he is handling COVID and the vaccine roll out.

Cluster 3 is again more vague but the event seems to be about the British public protesting lockdown and comparing it to Sweden one of the countries that did not implement a full lockdown.

Cluster 4’s event must be about the school going back due to the frequent use of ‘teacher’ and how parents must feel about this situation.



**b. Investigate the geo-localisation of events**