DATA AGGREGATION, BIG DATA ANALYSIS AND VISUALIZATION

1. Tejas Dhrangadharia (tejassha)
2. Karan Nisar (karankir)

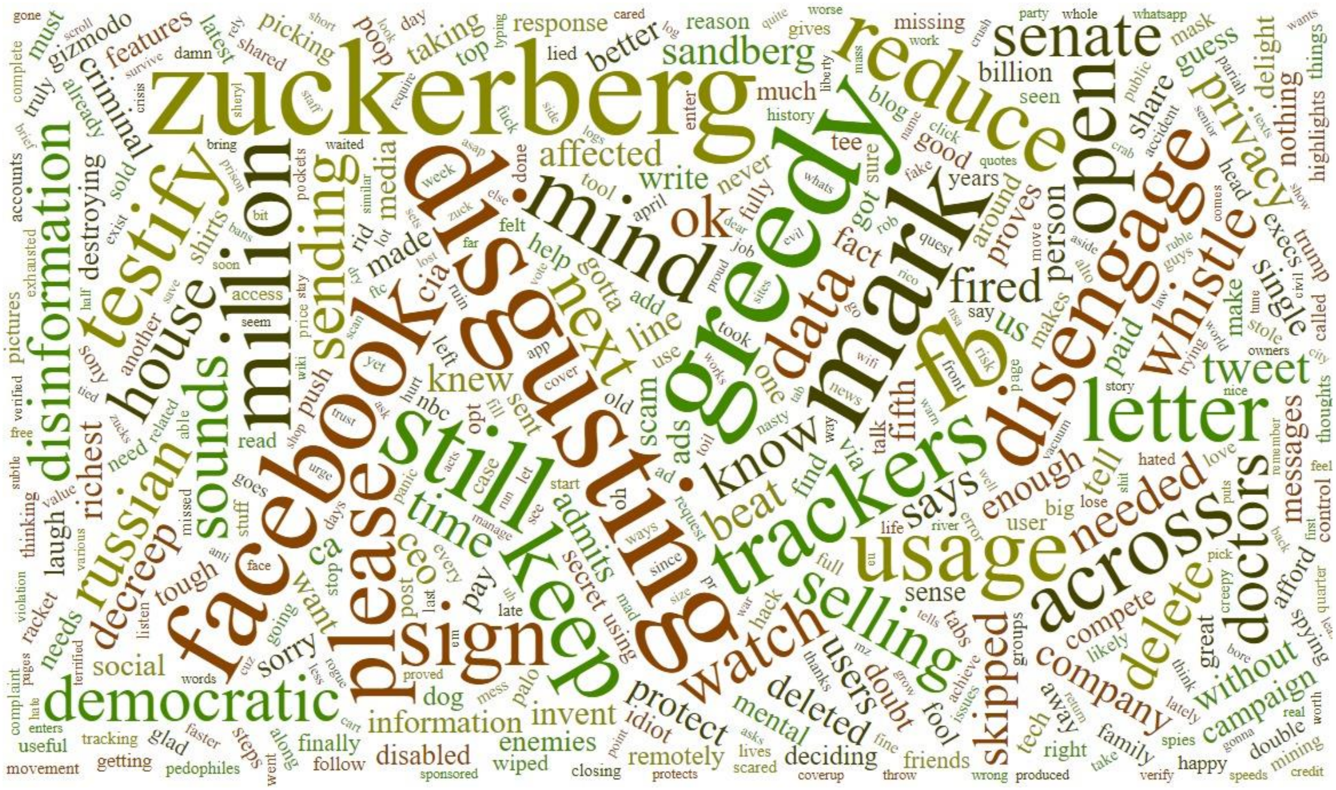
Our topic is the recent data breach involving Cambridge Analytica. This is a trending topic in the news currently. The trending twitter hash tag related to this incident is *#deletefacebook*. So, we collected tweets based on this hashtag on a daily basis starting from the day this hash tag became trending. We used the NYTimes API to collect articles on Facebook. However, using NYTimes API we could only obtain a snippet of articles and not the entire articles, we collected the urls of the articles and wrote our scraper to extract the entire article data. We collected a total of over 20000 tweets and 150 articles of NYTimes.

We stored this data into 2 directories: TwitterData and NewsData.

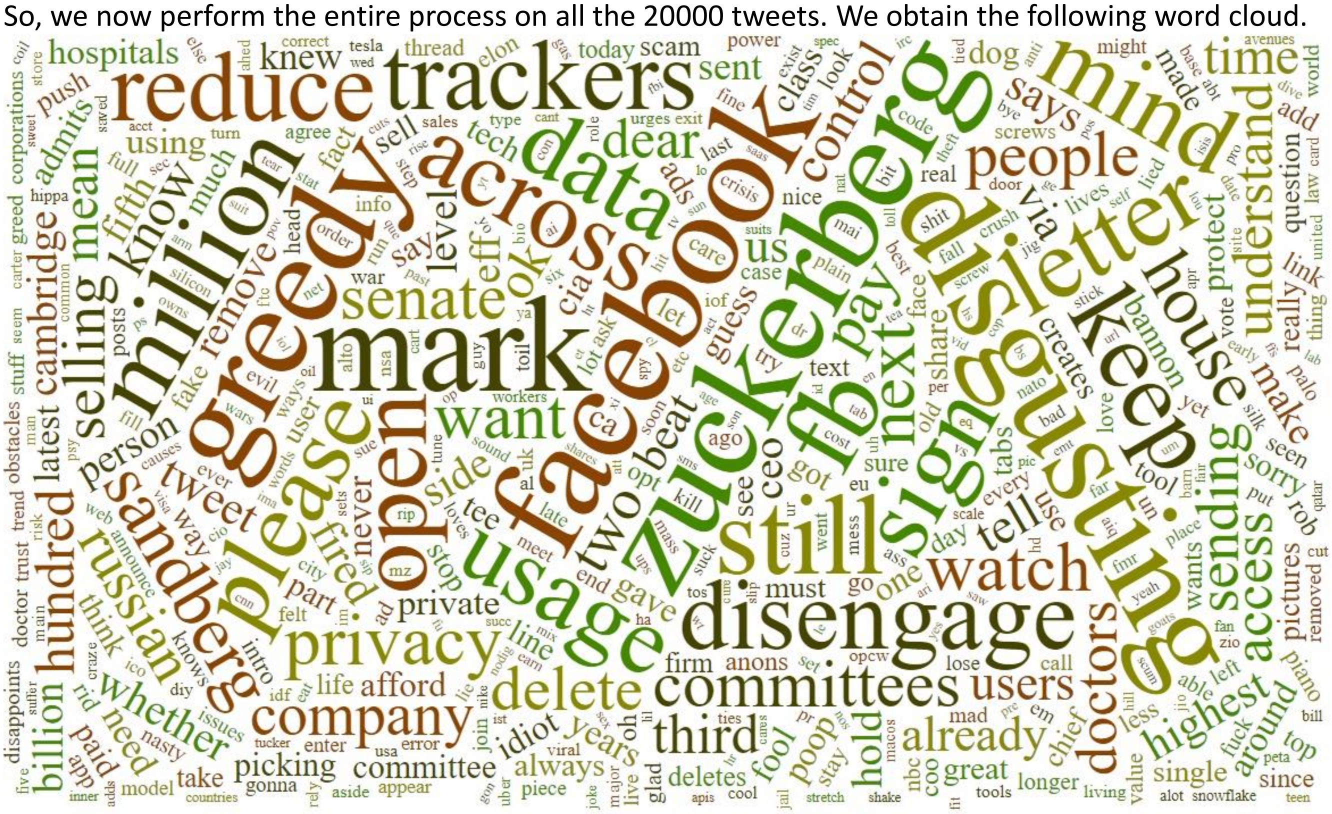
Now, we will perform Map and Reduce to obtain Word Count and Co-Occurrence of words. The First step in mapper would be cleaning the data and removing stop words. We used Python library nltk for this. The data is split in words, the words are converted to lower case and English stopwords present in the nltk library are removed.

This clean dataset of words are processed in the mapper to obtain a key-value value pair of all the words which will be processed in the reducer. The reducer will group all the keys and add the values of all those keys to find the occurrences of a particular word.
Once we have the occurrences we created a word cloud and displayed it using d3.js on a simple web page. However here we notice that a lot of irrelevant words are present in the data which are not needed and have high frequency of occurrences. Some of those words are ‘said’, ‘could’, etc. So we again eliminate these words individually in our cleaning phase of mapper so that we only have relevant words.

Once again, the same process is repeated and the mapper and reducer were implemented. We got the following wordcloud for a small sample of 1000 tweets data.



We see that the words Zuckerberg, facebook, mark have high frequency which makes sense.
So, we now perform the entire process on all the 20000 tweets. We obtain the following word cloud.



We see that the words with highest occurrence are almost the same. So, our output converges here.
We can also see convergence for in the word cloud of NY time data.

We have displayed the outputs on the web page we created using d3.js.

Word Co-occurrence

Only obtaining the frequencies of individual words isn't enough. We can't obtain much insight from it. So, we drill deeper in our analysis and obtain the word co-occurrences.

Word Co-occurrence is finding the pair of words that have occurred the most together in a tweet for twitter data and in a paragraph in case of NY time data. We used Twitter sample data and News sample data for this. First task is to obtain the top 10 occurring words as these are the words of utmost importance to us. So we use the output of the reducer and obtain the top 10 words and then find the co-occurrence of these words in the dataset.

For obtaining co-occurrence we initially took each of the top 10 words and then calculated the frequency of that word in the dataset. But we realized that for a larger dataset this would not be efficient method as we are scanning the dataset for ten times which would result in a significant drop in performance. To overcome this problem, we scanned through the dataset only once and checked for occurrences of all top 10 words all at once. This will be efficient method in running over huge datasets.
The mapper gave the set of word and the co-occurring word <word, co-occurring word> as output which was collated by the reducer giving the co-occurrences of the top 10 words.

A snippet of the output of reducer is:

```
facebook-zuckerberg 5
facebook-mark 3
mark-zuckerberg 3
facebook-zuckerberg 3
facebook-million 14
```

We than visualize this output on wordcloud using d3.js.