Master Group Assignment

SOCIAL MEDIA ANALYTICS - BASIC And OFFENSIVE TEXT ANALYSIS

Melissa Ho Hui Qi, Jiayun DENG, Xi LUO
School of Computer Science, University of Glasgow,
Glasgow, UK
Email: [2442814M, 2488598D, 2495109L]@student.gla.ac.uk

Source Code: https://github.com/melissaho97/social_media_analytics.git

***Abstract* - Social Media Analytics is a popular research area which is used to retrieve useful information from social media platforms, i.e Twitter. This paper introduces how Twitter data is collected using Twitter API and uploaded into MongoDB. Later, these crawled data will later undergo data analysis to obtain the statistics of the data collected, geo-tagged data from London, redundant data within the collection, number of re-Tweets and quotes, as well as the multimedia content types. Furthermore, these collected data will undergo further analysis in detecting offense, aggression, and hate speech contained within the Tweets alongwith the help of hate and offensive language databases.**

Index Terms - Social Data Analytics, Twitter; MongoDB; GitHub; Data Crawling; Basic Data Analytics, Offensive Language, Hate Speech

I. INTRODUCTION

In recent days, social media platforms have been popular for everyone to post and share their own thoughts and knowledge with the rest of the world. Over the years, more social media platforms have been established and gained popularity within the society. For example, MySpace, Facebook, Twitter, Tumblr, Instagram. Hence, the postings of the society or known as Tweets in the context of Twitter will later then contribute and become a Big Data which could be analysed to retrieve useful information from it. This process is also known as Social Media Analytics. Thus, this paper will basically describe the process of Social Media Analytics in the context of Twitter as the selected social media platform. At the beginning of the paper, it will first address how data is crawled and collected along with the help of Twitter API. In addition to it, one has to register him/herself within Twitter Developer Account before being able to use the Twitter API in order to obtain the required API Key, API Secret Key, Access Token, and Access Secret Token. Later, this paper will also describe APIs used for obtaining data through Twitter Streaming API, hybrid architecture of Twitter Streaming and REST APIs as well as obtaining geo-tagged data for London. Furthermore, this paper will elaborate in details regarding the process of basic data analytics in order to obtain and calculate the amount of data collected, geo-tagged data from London, redundant data present in the collection, re-tweets and quotes, as well as the multimedia content types.

Source Code

For this assignment, we have referred to several articles and examples of Python code online which guided us step-by-step regarding how to crawl data through Twitter API as well as how to analyse the obtained data in order to retrieve useful information from it.

For instance, with referencing to the source **[3], [4]** and **[5]**, we have known how to register for Twitter Developer Account in order to obtain API Key, API Secret Key, Access Token and Access Secret Token which will later to be added within the Python code as the authorisation permission for crawling the Tweets data from Twitter. Within the code, we have also included and implemented the use of Twitter API, Tweepy and StreamListener as our data crawling method.

After obtaining the Tweets data from Twitter, we also refer to the guidelines stated within the source **[1]** and **[8]** to register and create a cluster as well as default user account through MongoDB website besides uploading the obtained data and track the data changes through MongoDB Compass software for further data analysis. The following is our MongoDB Cluster link: mongodb+srv://user_readonly_WS:webscience@clusterws-qwtte.mongodb.net/test

Furthermore, we have uploaded our code onto our GitHub repository: https://github.com/melissaho97/social_media_analytics.git.

Time and Duration of Data Collected

According to the source **[9]**, 8am to 9am is the peak hours for Europeans to be active on Twitter. Hence, we have decided to analyse the data on 9th February 2020 starting from GMT 08:00 to GMT 09:00. The reason why we have selected particularly on 9th February 2020 due to a lot of things happening on this date. For instance, it is one day after Chinese New Year celebration, the worst day for Storm Ciara to hit the UK **[14]**, National Pizza Day **[13]**, and others more.

II. DATA CRAWLING

Data Crawling is the process of retrieving data from the defined data source, i.e. Twitter social media platform.

Twitter Streaming API

As mentioned in the previous section, we have used Twitter API and Tweepy in order to crawl 1% Tweets data from Twitter along with the guidelines from the source **[3], [4]** and **[5]**.

Hybrid Architecture of Twitter Streaming and REST APIs

REST stands for Representational State Transfer protocol, which is commonly used at the client server side. **[10]**

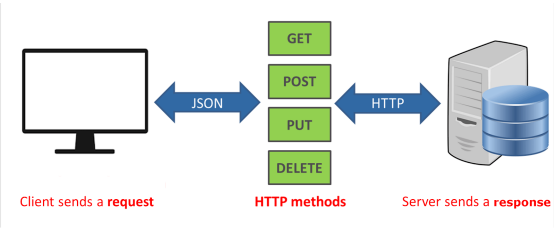

Figure 1: REST API Architecture **[10]**

For this, we have implemented StreamListener which complies with the concept of REST APIs along with Twitter API and Tweepy in order to retrieve the data. Regarding this, we included the time duration which we intended to search (9th February 2020 starting from GMT 08:00 to GMT 09:00) within our Python code as one of the searching criteria.

Geo-tagged Data for London Area

Within the code, using the StreamListener, we have entered the keywords as well as defined the area scope for London, UK as its additional searching criteria besides the defined search time (starting from 9th February 2020 GMT 08:00 to 9th February 2020 GMT 09:00).

Data Access Strategies and Restrictions

Throughout the process, we also acknowledge the data access strategies and restrictions on crawling data from Twitter. To address these, we have registered one Twitter Developer Account which could be shared and used within the group. With the grant access within the Twitter Developer Account, we managed to obtain the API Key, API Secret Key, Access Token, and Access Secret Token as our authorised access in retrieving the Tweets data using the Twitter API.

Our authorised access are as the following:

```
API Key:
aL2vSpF28tT9N1LKYqrGiIPwh
API Secret Key:
Zp2GqjePvVX3mpRB8Ts4nj07KP0KDfxND
TPgoP8nf7BA3uO9RR
Access Token:
1222856641877610502-A24IkwTQuUYqM
iBauBCzsF7qjmkeFc
Access Secret Token:
T8PaQ5dXkTGBsPKeHOLtdsBMUZ3WmDzp6
08KXTBMs0jna
```

III. BASIC DATA ANALYTICS

Using the Twitter API, Tweepy and StreamListener, we have managed to obtain our desired result as the following.

Amount of Data Collected

Figure 2 shows the total amount of data which has been retrieved within the planned time frame(starting from 9th February 2020 GMT 08:00 to 9th February 2020 GMT 09:00).

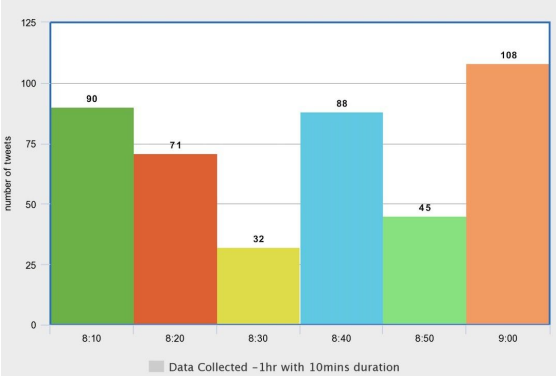

Figure 2: Histogram of the Amount of Data Collected over 10 mins periods

Amount of Geo-tagged Data from London

Figure 3 shows the total amount of geo-tagged data which has been retrieved within the planned time frame(starting from 9th February 2020 GMT 08:00 to 9th February 2020 GMT 09:00) and of the specific location: London.

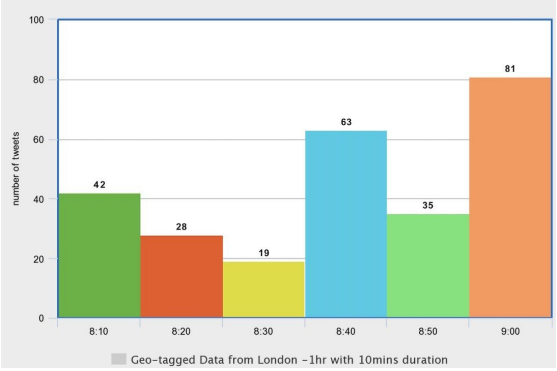

Figure 3: Histogram of the Amount of Geo-tagged Data from London over 10 mins periods

Amount of Redundant Data Present in Collection

Figure 4 shows the total amount of data which has been retrieved within the planned time frame(starting from 9th February 2020 GMT 08:00 to 9th February 2020 GMT 09:00).

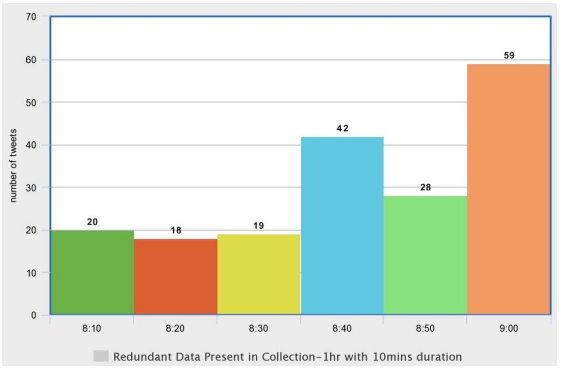

Figure 4: Histogram of the Amount of Redundant Data Present over 10 mins periods

Amount of Retweets and Quotes

Figure 5 shows the total amount of data retweets and quotes which have been retrieved within the planned time frame(starting from 9th February 2020 GMT 08:00 to 9th February 2020 GMT 09:00).

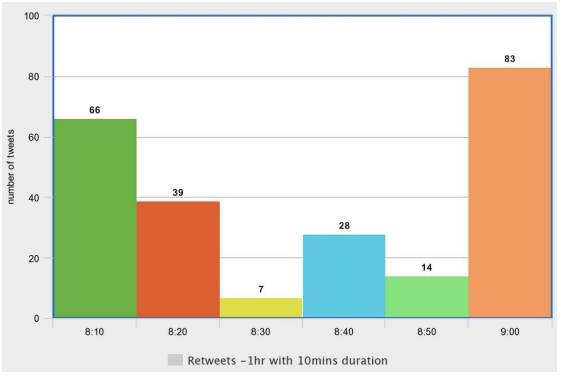

Figure 5: Histogram of the Amount of Retweets and Quotes over 10 mins periods

Amount of Multimedia Content Types

Figure 6 shows the total amount of data types collected within the planned time frame(starting from 9th February 2020 GMT 08:00 to 9th February 2020 GMT 09:00)

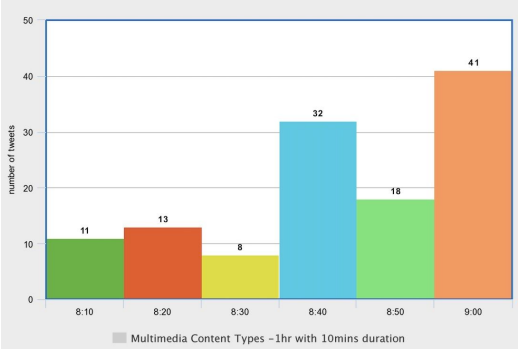

Figure 6: Histogram of the Amount of Multimedia Content Types over 10 mins periods

IV. ADVANCED DATA ANALYTICS

This section will describe the process on how we analyse the collected Tweets data whether it contains offensive, aggressive or hate speech by comparing alongwith the data dictionary of offensive and hate speech contributed by the sources of **[11]** and **[12]**.

Background and Motivation

The purpose of this selected task is to avoid abusive usage of Social Media platforms.

Solution

Using the obtained data (starting from 9th February 2020 GMT 08:00 to 9th February 2020 GMT 09:00), we have identified a total of 165 offensive words within the collected Tweets data.

Later, we categorised the offensive words according to their offensive types based on the defined types within the offensive dictionary. The following table is an extract of the top 5 offensive word types which are found within the Tweets data.

#	Offensive Word	Number of Tweets
1.	bullshit	42
2.	fuck	35
3.	shit	30
4.	bitch	23
5.	nigga	22

Table 1: Top 5 Offensive Word Types found in Tweets Data

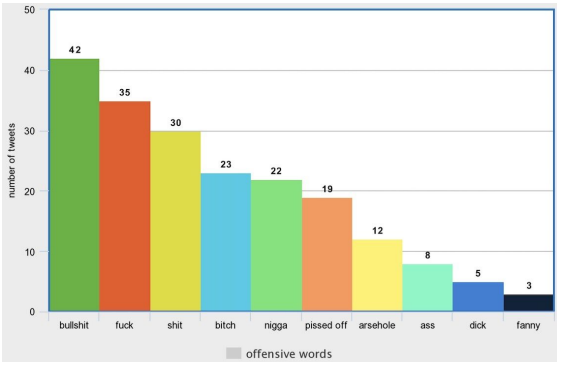

Figure 7: Bar Chart of Top 10 Offensive Words Found in Tweets Data

Discussion

According to Table 1 and Figure 7, it can be concluded that the top 5 offensive words which are commonly used in Tweets are 'bullshit', 'fuck', 'shit', 'bitch', and 'nigga'. It can be seen from the collected Tweets that offensive words have been commonly used within the era of social networking.

V. CONCLUSION

For this assignment, our team has done streaming tweets using twitter API - tweepy with filtering the location of the tweets in London, United Kingdom with geolocation and also limiting the streaming for one hour of a random day and geo-coded data. Connect the mongoDB cluster we created for the project and import data into it. Then, the data will detect offensive language, aggression and hate speech in twitter tweets. Also, for the sub-tasks the team identifies offensive language in tweets, categorising offensive types with providing the statistics of crawled tweets. For improvement and possible optimization for the project is that if the team has more time to complete the work, the team could also do research on the other three options to learn more about Twitter crawler for data collection with meme emotion analysis, noisy named entity recognition, and visual event summarisation. After finishing the project, all members from the team have a better acknowledgement with web science with twitter crawler and the usage of twitter APIs.

Acknowledgement

We would like first to express our deep and sincere gratitude to our lecturer, Professor Joemon Jose, Professor for Web Science module, University of Glasgow, for providing us his invaluable guidance

and support in ensuring the whole project to work out smoothly. This has made the whole workload being possible completed along with a great explanation of how the assessment should be like and supporting related material which are being provided in classes. Moreover, the deadline extension of the project also helps the team a lot by giving us more time to complete and polish up our project work, enabling it to perform in a better condition with optimized results.

Besides that, we are thankful to work as a team for this project. Although we have a hard time getting along with one another, it is a miracle that we are able to work out this project as one.

Many thanks to our friends as well who have guided and supported us throughout the project endlessly.

References

1. Import Data into MongoDB: 2020. *https://docs.mongodb.com/guides/server/import/*. Accessed: 2020- 03- 20.
2. How To Track Tweets by Geographic Location: 2020. *https://www.bmc.com/blogs/track-tweets-location/*. Accessed: 2020- 03- 20.
3. API Reference — tweepy 3.5.0 documentation: 2020. *http://docs.tweepy.org/en/v3.5.0/api.html*. Accessed: 2020- 03- 20.
4. Jayasekara, D. 2019. Extracting Twitter Data, Pre-Processing and Sentiment Analysis using Python 3.0. Medium. https://towardsdatascience.com/extracting-twitter-data-pre-processing-and-sentiment-analysis-using-python-3-0-7192bd8b47cf.
5. Sistilli, A. 2017. Twitter Data Mining: A Guide to Big Data Analytics Using Python. Toptal Engineering Blog. https://www.toptal.com/python/twitter-data-mining-using-python.
6. Connection String URI Format — MongoDB Manual: 2020. *https://docs.mongodb.com/manual/reference/connection-string/*. Accessed: 2020- 03- 20.
7. Python API Tutorial: Working with Streaming Twitter Data: 2020. *https://www.dataquest.io/blog/streaming-data-python/*. Accessed: 2020- 03- 20.
8. Joseph, M. and Wasser, L. Work With Twitter Social Media Data in Python - An Introduction. Earth Data Science - Earth Lab. https://www.earthdatascience.org/courses/earth-analytics/using-apis-natural-language-processing-twitter/intro-to-social-media-text-mining-python/.
9. Lee, K. 2016. The Best Time to Tweet & Why | Buffer Blog. Resources. https://buffer.com/resources/best-time-to-tweet-research.
10. Benharosh, J. 2018. What is REST API | PHPenthusiast. Phpenthusiast.com. https://phpenthusiast.com/blog/what-is-rest-api.
11. Davidson, T., Warmsley, D., Macy, M., & Weber, I. 2017. Automated hate speech detection and the problem of offensive language. In Eleventh international aaai conference on web and social media.
12. Davidson, T. 2017. Hate Speech and Offensive Language. data.world. https://data.world/thomasrdavidson/hate-speech-and-offensive-language.
13. National Pizza Day - UK. Wincalendar.com. https://www.wincalendar.com/uk/Pizza-Day.
14. Badshah, N., Quinn, B. and Pidd, H. 2020. Storm Ciara: travel chaos and floods amid warning of 'danger to life' – as it happened. the Guardian. https://www.theguardian.com/uk-news/live/2020/feb/09/storm-ciara-travel-chaos-flood-wind-rain-danger-uk-weather-live-latest-updates.

GitHub Source Code

https://github.com/melissaho97/social_media_analytics.git

MongoDB Cluster

mongodb+srv://user_readonly_WS:webscience@clusterws-qwtte.mongodb.net/test