CSCM35: Big Data and Data Mining
Coursework 2

Andy Gray
445348

01/05/20

Swansea University
Prifysgol Abertawe

1 Introduction

We have presented upon us a challenge to complete a practical data mining task, which involves a technical report and a software solution. We have a dataset provided that has been collect by Johns Hopkins CSSE in real-time, with values related to the current pandemic the coronavirus (COVID-19). However, we need to use additional datasets to complete complement the provided data. Once identifying a research question, we need to develop a prototype and evaluate our method.

With the COVID-19 first declaring over 800 cases on the 23rd of January 2020, and figures still rising, which at point of writing was 3.12m cases and 217K deaths worldwide [?], the virus has reached over 199 different countries. We wanted to see if between the time scale of the 22nd of January and the 25th of April, what is the general public sentiment is. We want to know, has the general public sentiment changed, or if it has stayed the same? While over the COVID-19 pandemic.

We will first achieve this by using Twitters API to gain all the tweets that have been tweeted, over this time, using the hashtag 'coronavirus'. We will then complete a sentiment analysis on the data, to gain an overall view of the general feel of the 'coronavirus' tweets. Gaining the sentiment value will allow us to have insights into the public view, and if these tweets are positive, neutral or negative and how they change as the pandemic plays out over time.

The sentiment analysis showed that at the beginning of the declaration of the pandemic, Twitter's tweets had a slight positive sentiment at the beginning, then quickly declined into a negative sentiment. March had a very positive start to the month, then had a massive drop to negative figures in the middle of the month to return positive to then return slightly in the negative numbers at the end of the month. While April

The report will first explain our proposed solution. Explaining how we aim to get the tweets from Twitter, and then put the contents into a CSV file. We then will explain how we intend to use the Natural Language Toolkit (NLTK) to train our classification models. We will then explain how we intend to provide the classifiers with the extracted tweets and gain a sentiment score based on the content of the tweets. We will then explain the packages used and describe the dataset and any preprocessing techniques we used. We will then explain the results which are then followed by a discussion on the results and a conclusion.

2 Proposed Solution

We will be using twarc, a Python library for archiving Twitter JSON data. The library accesses the Twitter API to extract a JSON object as Twitter stores tweets as line-oriented JSON. The twarc library handles the Twitters's API, as well as its rate limits. This library also allows us to use hydrate tweet ids. It is collecting the relevant tweet's details that match the criteria of having hashtags of 'coronavirus' [?].

We will then use another python library called hydrate. Hydrate allows us to be able to take the Tweet ID's that we have taken off twitter, by using Twarc and get the contents of the actual tweet. Hydrate will provide us back with the content of the tweet, along with other potentially crucial information. These different parts of the tweets information we get are "created_at", "id", "id_str", "full_text", "source", "truncated", "in_reply_to_status_id", "in_reply_to_status_id_str", "in_reply_to_user_id", "in_reply_to_user_id_str", "in_reply_to_screen_name", "user", "coordinates", "place", "quoted_status_id", "quoted_status_id_str", "is_quote_status", "quoted_status", "retweeted_status", "quote_count", "reply_count", "retweet_count", "favorite_count", "entities", "extended_entities", "favorited", "retweeted", "possibly_sensitive", "filter_level", "lang", "matching_rules", "current_user_retweet", "scopes", "withheld_copyright", "withheld_in_countries", "withheld_scope", "geo", "contributors", "display_text_range", "quoted_status_permalink". We only need for this sentiment analysis the values "created_at" and "full_text". Once the tweets get collected for all of the Tweet IDs, we will then convert the JSON data into a CSV file. We will then do sentiment analysis on the tweets to gauge the over feel of these tweets. To see if they are all negative or if they overall a positive vibe over each day.

We will then use NLTK to create a sentiment analysis on the tweets text body. However, before we could do this, we needed to use several packages to get a trained model, to provide us with the likelihood of the tweet being positive or negative. To classify the tweet, we used SKLearn's Naive Bayes MultinomialNB and BernoulliNB, Linear Model's LogisticRegression, SGDClassifier and SVM's SVC, LinearSVC, NuSVC.

We then create a sentiment function, which takes in a parameter of text. This function will find the features of the parameter and then return the classification of the tweet if it was positive or negative, and the confidence level of that tweet with its analysis. The value we pass the function is the tweets 'full_text' value. The classification and confidence level gets outputted as a .txt file.

To visualise the data, we use Matplotlib and the outputted .txt to populate the graph. We are displaying the sentiment values of the tweets, allowing us to see the trend in the tweets sentiment analysis.

2.1 Packages

We used Python 3 [10] as our programming language as it allowed us to use the required libraries and additional pages we needed. One of the main libraries was NLTK, a leading platform for building programs to work with human language [?]. SKLearn [3] for its Naive Bayes, linear model and SVM libraries. Along with Statistics for its mode library. Altair and Matplotlib were used to visualise the data.

2.2 Dataset and Preprocessing

The presented dataset provided, which is maintained by Johns Hopkins CSSE, consists of [decsribe data here]. The data had many missing values, which correlate to the country not having any COVID-19 cases, so these features got filled in with zeros.

With using Twarc, we were able to gain 23.462 million tweets. Twarc was able to pull 1.857 million tweets between 22/01/20 to 28/02/20, 9.367 million between 01/03/20 to 31/03/20 and 12.238 million unique tweets between 01/04/20 to 25/04/20.

Through using Twarc and Hydrate, we had to drop features from the table that were not required. The required features we needed were 'created_at' and 'full_text'.

In order to train our sentiment analysis models, we used a sample .txt file that contained positive text, and another .txt file that contained negative text. Along with NLTK's class FreqDist, we were able to encode the frequency distributions. The frequency distribution counts the number of times an experiment has that outcome occur [?].

2.3 Sentiment Analysis

Sentiment analysis is a type of data mining. Sentiment analysis is a form of semi-supervised learning [?] which uses classification, a text-based classification approach [?]. It aims to measure what the inclination of people's opinions is, through using natural language processing (NLP). It can also get referred to as Society text data mining [?]. NLP is a computational linguistics and text analysis. It gets used to extract and analyse information from the Web, which is mostly social media and other similar sources. The analysed data quantifies the general public's sentiments or reactions toward certain situations, products, people or ideas and reveal the contextual polarity of the information. Sentiment analysis is also known as opinion mining [?].

Sentiment analysis can fall into two categories, pure statistics or a mix of statistics and linguistics [?]. Pure statistics use algorithms like the Bag of Words (BOW). This kind of algorithms filters the text down to only the words that the algorithm believes to have sentiment, taking into account no context to the sentence at all. Such models do not aim to understand the language, only analyse the statistical measures to classify the text.

The mix of statistics and linguistics approach uses an array of Natural Language Processing (NLP) techniques, along with statistics to allow the machine to understand the language truly. The algorithms achieve this by incorporating languages grammar principles into analyst of the text.

There are broadly two main outputs to sentiment analysis. One type of sentiment analysis output gets referred to as Categorical/Polarity. What this means is that the text will get classed as either positive, negative or neutral overall. While the other is Scalar/Degree. What this means is that a score is given based on a predefined scale, that ranges from highly positive to highly negative. This type of sentiment analysis output has been used on tweets to see the views on various USA election candidates.

3 Results

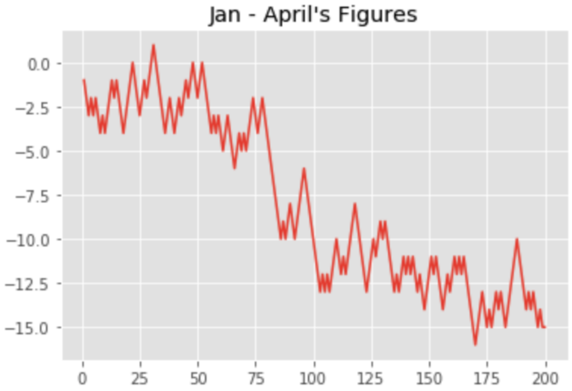

Fig: These graphs show the sentiment analysis for January 23rd to April 25th.

The results show that a negative trend has been happening since the 23rd of January. Although the overall values have been going in a negative direction, there has been occurrences where the overall tone has been more positive. However, not enough to take the value back over zero and into the positive figures. The highest gaining 2 which the lowest value being -15.5.

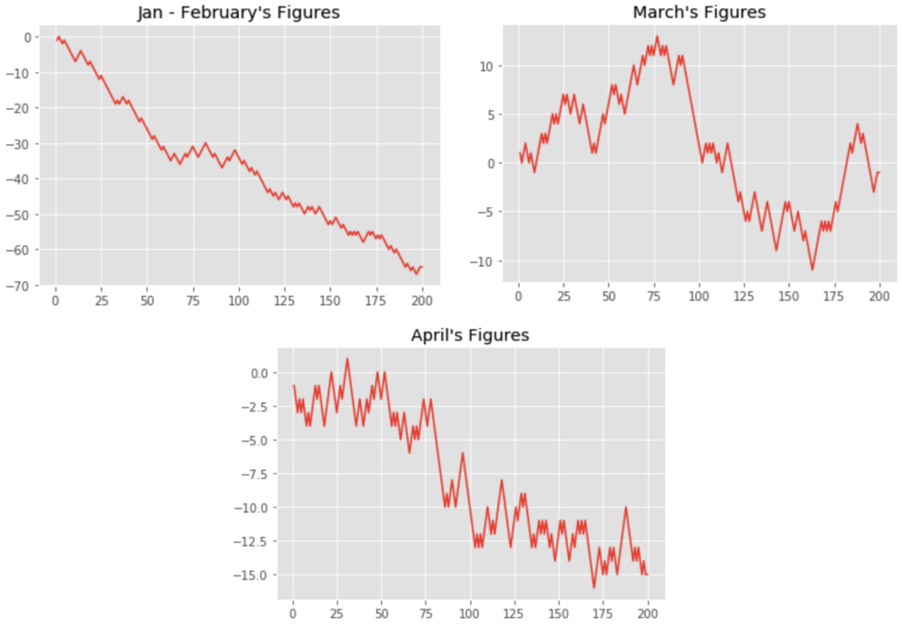

Fig: These show each months sentiment analysis individually.

As we can see from January to February **??** tweet sentiment analysis table, the values start to go in a positive direction but quickly move down into a negative position. With the occasional spike in the tweets positivity, however, not enough to stop the downward trend, ending the month on a value lower than -65.

On the other hand, March **??** had much more of an up and down result. In terms of the moods over the month, it went from extreme positive at the beginning to a sharp decline, to then return positive to go back slightly negative at the end of the month. The highest positive value was 15, while the lowest value was -12. It was showing that the month overall had more positive tweets than negative.

April's analysis shows that the month started in a negative place, but had a positive peak early on, to continue the downward trend then. With the highest positive value being 1.5 while the lowest value being -15.5.

4 Discussion

Understandably, January and February had a negative trajectory as this was the moment in time that the WHO was declaring the pandemic. Which naturally causes fear and confusion about what is in store for us with this virus. However, March started on a very positive trend, but that quickly started to go on a negative trajectory. It would be interesting to see what correlations can be made to what caused the positive start, could this be linked to the positive news coming out about people recovering and surviving? It is also interesting to see the steep negative trajectory as well, could this be linked to when most of the western worlds went into lockdown, as we have gained mainly English speaking nations tweets, who went into lockdown around the 23rd of March.

With the proposed solution only looking at words, a pandemic like we are currently in would have a lot of negative words get used which could result in the tweet getting seen as negative. However, if the context of the tweet gets taken into full account then the tweet itself might not be negative, just containing many words, that on their own and taken out of context, could be perceived as negative.

In regards to the sentiment analysis checking the tweets that contain the hashtag 'coronavirus', it does provide a limited scope. It would be good to be able to compare the results against other hashtags of a similar nature. For example, on the hashtags 'COVID-19', 'virus', [list more]. Have these hashtags overall had more of positive sentiment or even been used more due to the smaller size in word count?

5 Conclusion

References

[1] Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G. S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., Levenberg, J., Mané, D., Monga, R., Moore, S., Murray, D., Olah, C., Schuster, M., Shlens, J., Steiner, B., Sutskever, I., Talwar, K., Tucker, P., Vanhoucke, V., Vasudevan, V., Viégas, F., Vinyals, O., Warden, P., Wattenberg, M., Wicke, M., Yu, Y., and Zheng, X. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.

[2] Bradski, G. The OpenCV Library. *Dr. Dobb's Journal of Software Tools* (2000).

[3] Buitinck, L., Louppe, G., Blondel, M., Pedregosa, F., Mueller, A., Grisel, O., Niculae, V., Prettenhofer, P., Gramfort, A., Grobler, J., Layton, R., VanderPlas, J., Joly, A., Holt, B., and Varoquaux, G. API design for machine learning software: experiences from the scikit-learn project. In *ECML PKDD Workshop: Languages for Data Mining and Machine Learning* (2013), pp. 108–122.

[4] Hunter, J. D. Matplotlib: A 2d graphics environment. *Computing in science & engineering 9*, 3 (2007), 90–95.

[5] Krizhevsky, A., Sutskever, I., and Hinton, G. E. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems* (2012), pp. 1097–1105.

[6] LeCun, Y., Bengio, Y., and Hinton, G. Deep learning. *nature 521*, 7553 (2015), 436–444.

[7] LeCun, Y., Boser, B., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W., and Jackel, L. D. Backpropagation applied to handwritten zip code recognition. *Neural computation 1*, 4 (1989), 541–551.

[8] LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. Gradient-based learning applied to document recognition. *Proceedings of the IEEE 86*, 11 (1998), 2278–2324.

[9] LeCun, Y., et al. Generalization and network design strategies. *Connectionism in perspective 19* (1989), 143–155.

[10] Python Core Team. *Python: A dynamic, open source programming language.* Python Software Foundation, Vienna, Austria, 2020.

[11] Rumelhart, D. E., Hinton, G. E., and Williams, R. J. Learning internal representations by error propagation. Tech. rep., California Univ San Diego La Jolla Inst for Cognitive Science, 1985.

[12] Van der Walt, S., Schönberger, J. L., Nunez-Iglesias, J., Boulogne, F., Warner, J. D., Yager, N., Gouillart, E., and Yu, T. scikit-image: image processing in python. *PeerJ 2* (2014), e453.

[13] Walt, S. v. d., Colbert, S. C., and Varoquaux, G. The numpy array: a structure for efficient numerical computation. *Computing in Science & Engineering 13*, 2 (2011), 22–30.