https://doi.org/10.2352/EI.2023.35.3.MOBMU-355

Practical OSINT Investigation in Twitter Utilizing AI-based Aggressiveness Analysis

Artem Sklyar[1], Klaus Schwarz[1,3], Reiner Creutzburg[1,2]

[1]SRH Berlin University of Applied Sciences, Berlin School of Technology, Ernst-Reuter-Platz 10, D-10587 Berlin, Germany
Email: smalch1234@gmail.com, klaus.schwarz@srh.de, reiner.creutzburg@srh.de

[2]Technische Hochschule Brandenburg, Department of Informatics and Media, IT- and Media Forensics Lab, Magdeburger Str. 50, D-14770 Brandenburg, Germany
Email: creutzburg@th-brandenburg.de

[3]University of Granada, Faculty of Economics and Business, P.° de Cartuja, 7, ES-18011 Granada, Spain

Keywords: Open Source Intelligence, OSINT, SOCMINT, Reddit, Cybersecurity, Cyber Security, OSINT investigation, cybersecurity training

Abstract

Open-source intelligence is gaining popularity these days due to the development of social networks. There is more and more information in the public domain. Twitter is one of the most popular social networks, so it's worth analyzing its information. It was chosen to analyze the dependence of changes in the number of likes, reposts, quotes, and retweets on the aggressiveness of the post text for a separate profile since this information may be important not only for the owner of the channel on the social network but also for other studies that somehow affect user accounts and their behavior on the social network. also, the task of this work was a detailed analysis and evaluation of the capabilities of the tweety library and situations in which it can be effectively applied. also, the creation and description of a compiled neural network, the purpose of which is to predict changes in the number of likes, reposts, quotes, and retweets from the aggressiveness of the post text for a separate profile.

Introduction

Open-source intelligence (OSINT) is an intelligence discipline that synthesizes information from publicly available sources and analyzes it. In the intelligence community, the term “open source of intelligence data” indicates the public availability of the source (as opposed to secret sources and sources with limited use), but it is not related to the concepts of open source information, meaning any information in the media space.

Twitter is a USA microblogging service and social network where users post messages, known as “tweets”, and interact with them. Users interact with Twitter through a browser, a mobile app, or an API. Until April 2020, the services were available via SMS. The service is provided by Twitter, Inc., based in San Francisco, California, and has more than 25 offices worldwide. Initially, tweets were limited to 140 characters, but in November 2017, the limit was doubled to 280 for most languages. Audio and video replies remain limited to 140 seconds for most accounts.

The marketing research company Pear Analytics analyzed 2,000 tweets (information from the US and in English) for 2 weeks in August 2009 from 11:00 am to 5:00 pm (CST) and divided them into six categories. As shown in the Figure 1.[1]

1) News (Green)
2) Spam (Purple)
3) Self-promotion (Orange)
4) Small talk (Red)
5) Conversations (Blue)
6) Retweets or repeated messages (Brown)

Task description

Twitter is a popular social network, so keeping your account is essential for some people. An incorrectly written tweet can lead to severe ratings or monetary losses. But people are different, and everyone chooses the one he likes best. Therefore, it is very important to determine what the audience expects you to hear, or rather even to hear something, but how. The same news can be presented in entirely different ways. You might think that the less aggressive and positive a tweet sounds, the better. But this is not always the case.

Because of the unique structure of the human psyche, which was formed by evolution for survival in the wild, negative or aggressive emotions are remembered most of all. And the more aggressive they are, the better they are remembered. The next most effective memorization is positive emotions. Neutral ones are remembered very poorly.

But this does not mean that the more aggressive and negative your tweet sounds, the better. If people follow your news for positive content, for example, if you own a large company, such a tweet is likely to be perceived negatively, and your stocks will fall.

There are several metrics for users' evaluation of a particular tweet:

The number of likes is shown in Figure 2. This is the number of times users clicked to express what they liked.

The number of retweets is shown in Figure 3. This is how often users have shared (distributed) your post to others. This is what they do with those posts that they consider particularly important.

The number of replies is shown in Figure 4. You can interpret them as a kind of comment. How many times have users expressed their opinion about this tweet?

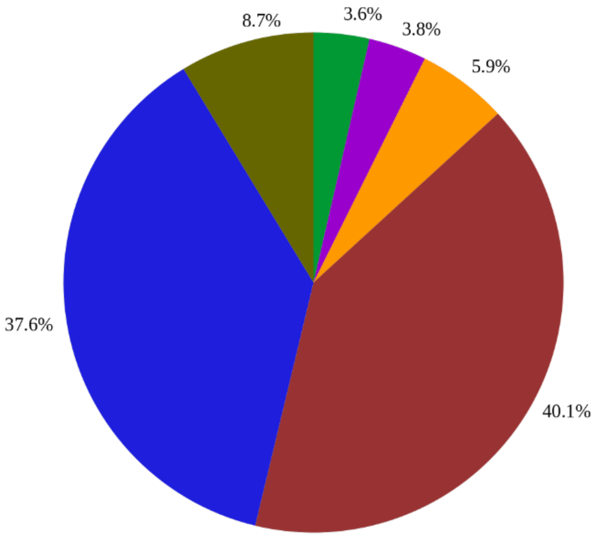
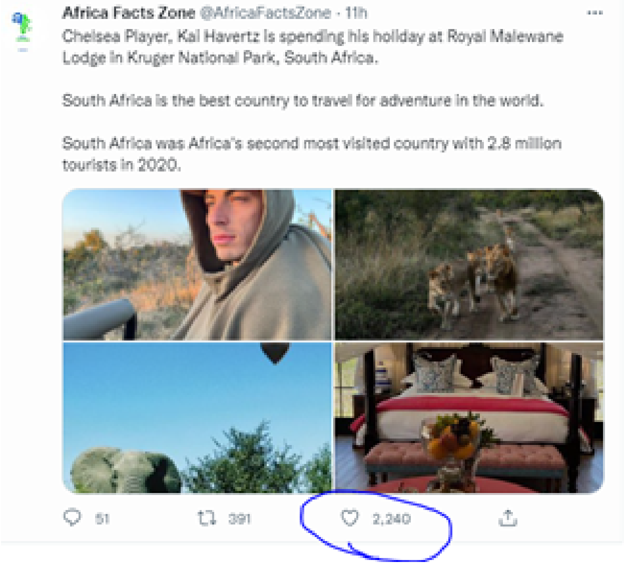


Figure 1. Six categories of Twitter Tweets [1]



Africa Facts Zone @AfricaFactsZone · 11h

Chelsea Player, Kai Havertz is spending his holiday at Royal Malewane Lodge in Kruger National Park, South Africa.

South Africa is the best country to travel for adventure in the world.

South Africa was Africa's second most visited country with 2.8 million tourists in 2020.

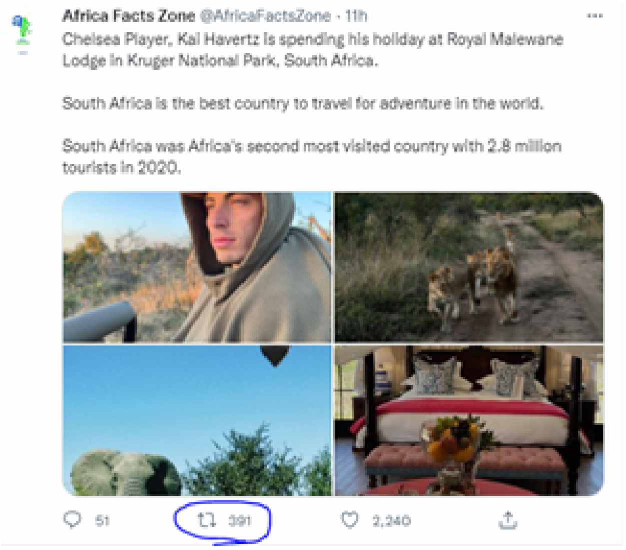
Figure 2. Likes from Twitter

And the number of quotes. A quote tweet is a retweet that contains some additional text.

All these four metrics reflect one or another audience engagement.

We will try to build a neural network that will use the history of recent tweets to analyze the audience's tastes. That is, we will be able to see how the audience will react to tweets with varying degrees of aggressiveness. the degree of reaction will be measured by changing the number of these four metrics.

The importance of such an analysis cannot be underestimated because if users do not like tweets, then, as already mentioned above, this will have negative consequences for the author.



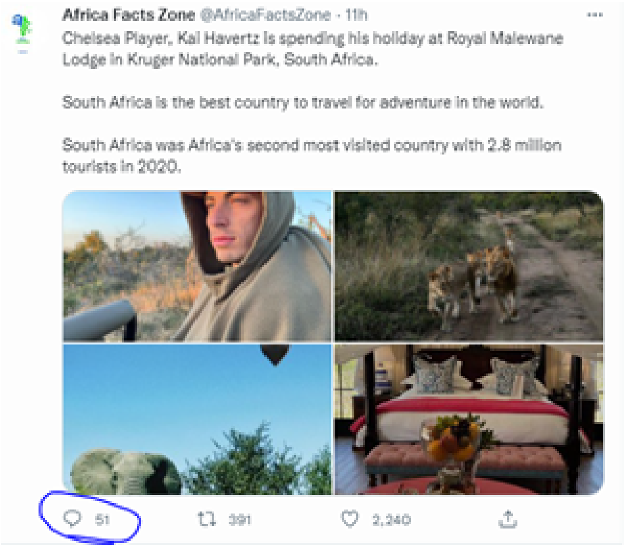
Africa Facts Zone @AfricaFactsZone · 11h

Chelsea Player, Kai Havertz is spending his holiday at Royal Malewane Lodge in Kruger National Park, South Africa.

South Africa is the best country to travel for adventure in the world.

South Africa was Africa's second most visited country with 2.8 million tourists in 2020.

Figure 3. Retweets from Twitter



Africa Facts Zone @AfricaFactsZone · 11h

Chelsea Player, Kai Havertz is spending his holiday at Royal Malewane Lodge in Kruger National Park, South Africa.

South Africa is the best country to travel for adventure in the world.

South Africa was Africa's second most visited country with 2.8 million tourists in 2020.

Figure 4. Replies from Twitter

Expected results

The graphs for likes, quotes, retweets, and replies are expected to be approximately the same since these metrics express people's interest in different forms.

It is also expected that representatives of companies will have the form shown in Figure 5. Here the aggressiveness of the text is marked on the x-axis. -1 - very aggressive. 1-positive. 0-neutral.

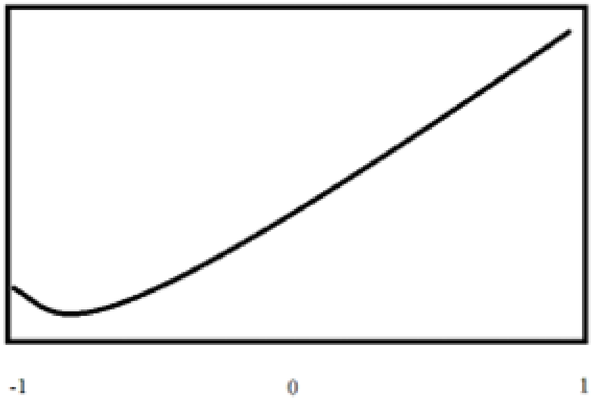


Figure 5. Expected graph for companies

This should be because it is beneficial for the company to tell everyone that things are going well, even if they are not.

For bloggers, the expected picture will be either the same as shown in Figure 6 or the same as in Figure 7, depending on the audience's bias.

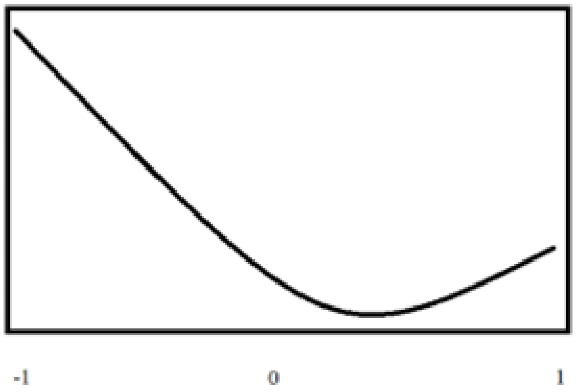


Figure 6. First of expected graph for bloggers

Used libraries

Currently, the actively supported library for interacting with Twitter is tweetpy.

But to be able to work with it, you need to register as a developer through your Twitter account.

There are 3 options. Their limits are mentioned in Figure 8.[2]

Tweetpy offers various methods for interaction. Most of them are really useful, but according to my search, some are useless. Let's try to value them.

Description of tweetpy methods and evaluation of their usefulness from 0 to 10:[3]

Description of situations in which the method data can be used. Based on it, a utility assessment was made for table 1.

TextBlob was used for semantic analysis.

Initially, the text should be cleared of various characters, such as / or brackets. After cleaning, you can calculate the polarity of the comment using a TextBlob for each comment separately.

Complete cleanup and run TexnBlob for semantic analysis are shown in Figure 9.

Theory

Simplistically, the work of the semantic analyzer can be represented as the scheme shown in Figure 10. [4]

Initially, we analyze the dataset and bring the text lengths to the same length by adding empty characters to the end of shorter ones. Next, we tokenize and delete the stop words. This can be represented by removing insignificant or insignificant words from sentences to shorten their

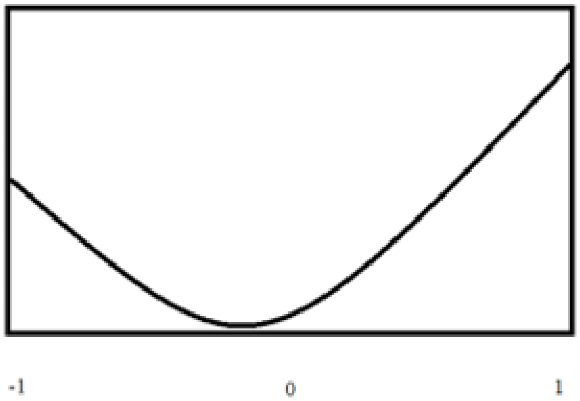
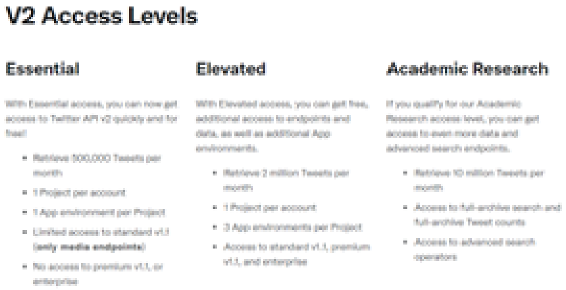


Figure 7. Second of expected graph for bloggers



V2 Access Levels

Essential

With Essential access, you can now get access to Twitter API v2 quickly and for free!

- Retrieve 500,000 Tweets per month
- 1 Project per account
- 1 App environment per Project
- Limited access to standard v1.1 **(only media endpoints)**
- No access to premium v1.1, or enterprise

Elevated

With Elevated access, you can get free, additional access to endpoints and data, as well as additional App environments.

- Retrieve 2 million Tweets per month
- 1 Project per account
- 3 App environments per Project
- Access to standard v1.1, premium v1.1, and enterprise

Academic Research

If you qualify for our Academic Research access level, you can get access to even more data and advanced search endpoints.

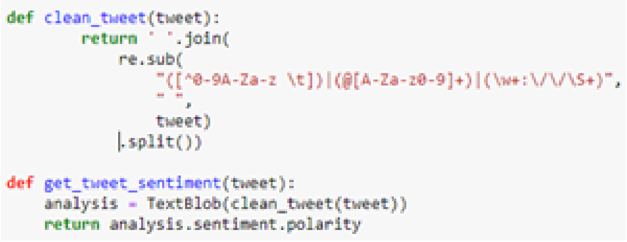
- Retrieve 10 million Tweets per month
- Access to full-archive search and full-archive Tweet counts
- Access to advanced search operators

Figure 8. Account limits [2]

length and not confuse the analyzer since they carry almost no meaning.

Next, we will transform them into code; since the algorithm does not understand the text, any words must be encoded. Next, we classify and evaluate.

A neural network is a very old concept, known back in the 20th century. But some aspects of it still require clarification. This is an activation function, namely, the difference of these activation functions applied after each neural network level.



```
def clean_tweet(tweet):
        return ' '.join(
            re.sub(
                "([^0-9A-Za-z \t])|(@[A-Za-z0-9]+)|(\w+:\/\/\S+)",
                " ",
                tweet)
            .split())

def get_tweet_sentiment(tweet):
    analysis = TextBlob(clean_tweet(tweet))
    return analysis.sentiment.polarity
```

Figure 9. Complete cleanup and semantic analysis

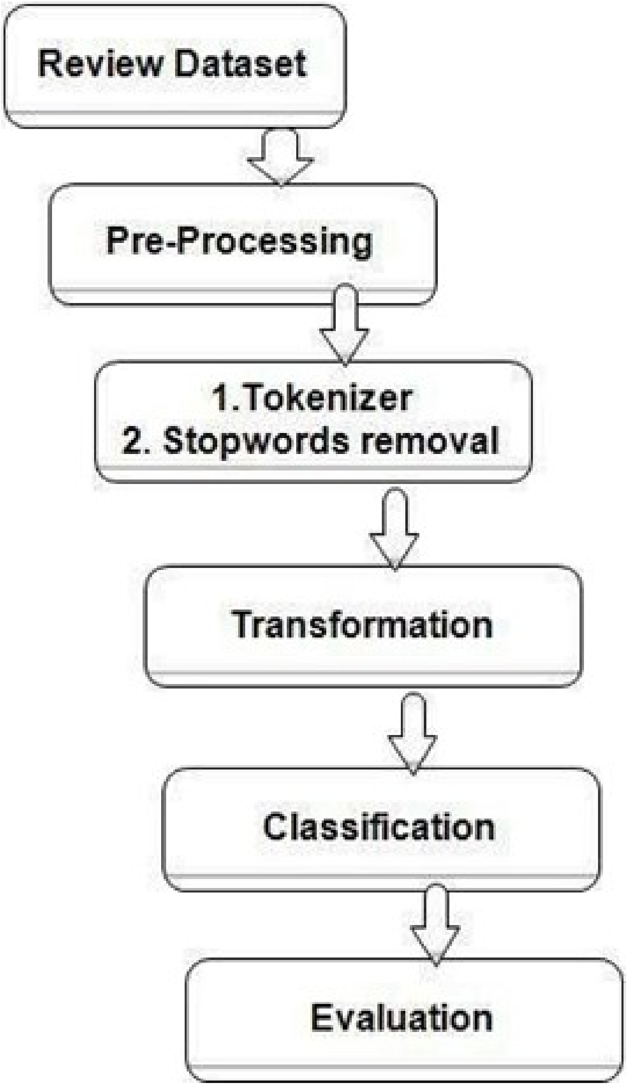


Figure 10. Semantic analyzer scheme [4]

The sigmoid function, shown in Figure 11, converts the incoming values into the real range [0, 1]. If the input data turns out to be largely positive values, then after the conversion, they will be approximately one, and negative numbers will become close to zero. This fairly popular function can be interpreted as the frequency of neuron excitation.[5]

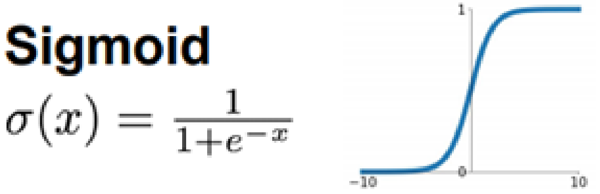


Figure 11. Sigmoid [5]

The tangent, shown in Figure 12, is very similar to the sigmoid, but it has two significant differences: it converts data to the range [-1, 1]. It has zero centering, which eliminates the second problem of the sigmoid. The gradient values can still be reset during reverse propagation; however, a tangent is usually preferable.

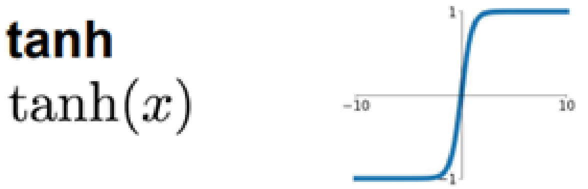


Figure 12. Tanh [5]

ReLU, or Rectified Linear Units, shown in Figure 13, has become quite popular recently. It calculates the function f(x) = max(0,x); that is, it simply outputs the values ”zero” and ”not zero”. This solves the problem of zeroing the gradient for positive numbers. In addition, the ReLU is very simple to calculate (about six times faster than the sigmoid and tangent).

That is why it was decided to choose this activation function to solve this problem.

LReLu is shown in Figure 14. The difference of this function is that it has a slight slope in the left half plane, which means that with harmful input data, the gradient will not be zero.

Maxout, shown in Figure 15, selects the maximum sum of two sets of weights multiplied by the original data, considering the offset. Thus, it generalizes Rely and leaky ReLU without zeroing the gradient. But, as you can guess by the type of function, maxout requires doubling the parameters and neurons.

Another concept that needs explanation is the concept of retraining and the concept of Dropout, which is a solution to the problem of retraining. In the learning process, all machine learning algorithms are prone to overfit. Conditionally, this can be understood as if the algorithm did not try to understand the rule by which it is possible to get answers (which is what we want from it) but simply remembered the answers. Overfitting is one of the problems of Deep Neural Networks. For example, a model explains only examples from the training sample well, adapting to the training examples instead of learning to classify examples that did not participate in the training (losing the ability to generalize). In recent years, many solutions to the problem of retraining have been proposed, but one of them has surpassed all the others thanks to its simplicity and excellent practical results. This solution is Dropout [6].

Figure 16 shows a neural network before applying Dropout, and Figure 17 shows a neural network after applying Dropout.

The main idea of Dropout is to train an ensemble of several Deep Neural Networks instead of training one and then take the average from the results.

Networks for training are obtained by excluding neurons from the network (dropping out) with a probability of p, so the probability that the neuron will remain in the network is q=1-p. The exclusion of a neuron means that for any input data or parameters, it returns 0.

Excluded neurons do not contribute to the learning process at any stage of the back-propagation algorithm; therefore, excluding at least one of the neurons is equivalent to training a new neural network.

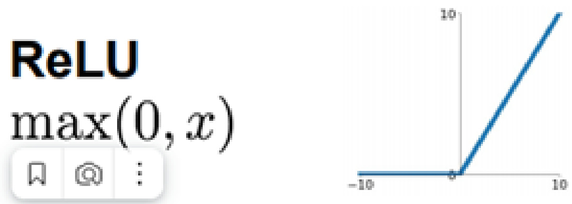


Figure 13. ReLu [5]

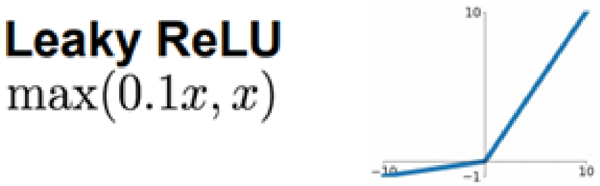


Figure 14. LReLu [5]

Maxout

$$\max(w_1^T x + b_1, w_2^T x + b_2)$$

Figure 15. Maxout [5]

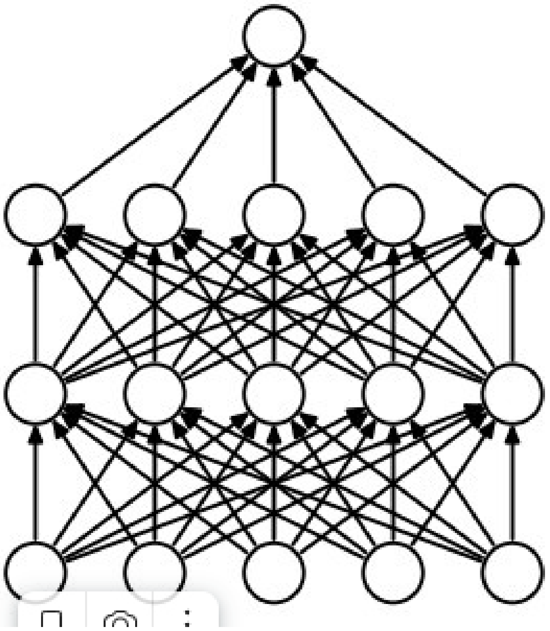


Figure 16. Neural Network before dropout [6]

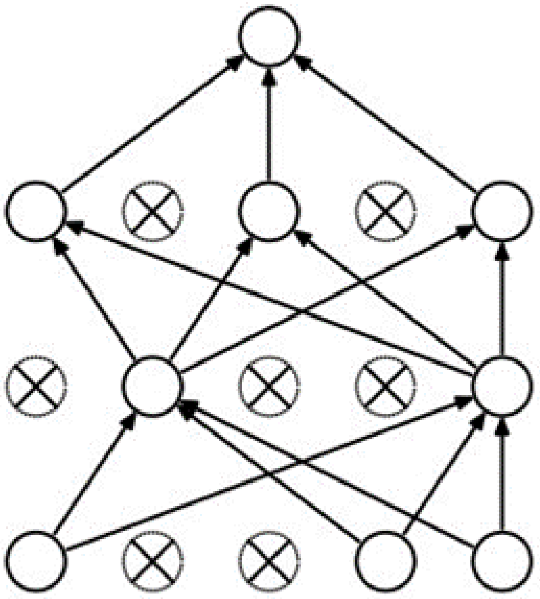
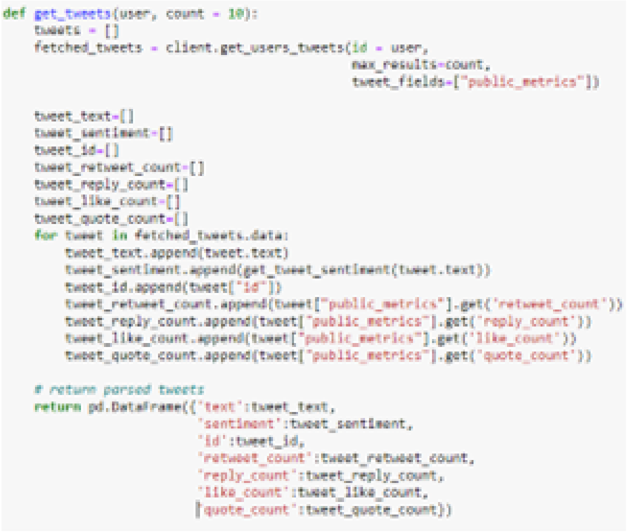


Figure 17. Neural Network after dropout [6]

Dataset

Initially, the algorithm receives the user's name as input; then, using the methods described above, it receives the user's ID, then his last comments with the metrics described above. The dataset itself then looks as shown in Figure 18.

Figure 19 shows a full python function using the user id and the get__users__tweets method with the public__metrics attribute to get information about metrics, too, to extract raw information. The result of the get__users__tweets method is later reformatted into a DataFrame, a regular table.

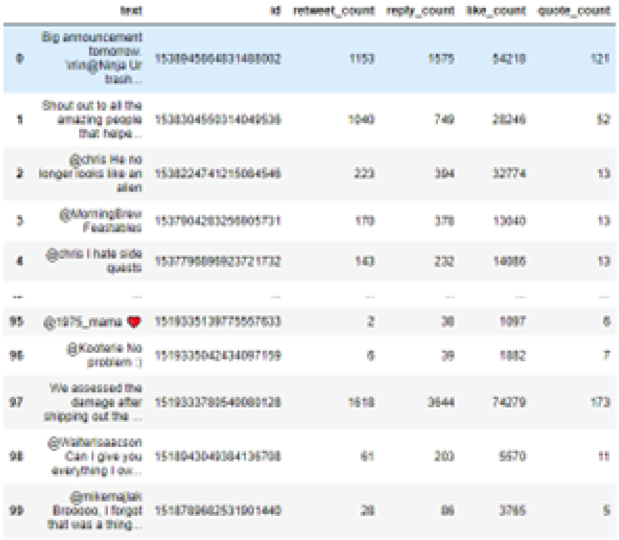


```
def get_tweets(user, count = 10):
    tweets = []
    fetched_tweets = client.get_users_tweets(id = user,
                                             max_results=count,
                                             tweet_fields=["public_metrics"])

    tweet_text=[]
    tweet_sentiment=[]
    tweet_id=[]
    tweet_retweet_count=[]
    tweet_reply_count=[]
    tweet_like_count=[]
    tweet_quote_count=[]
    for tweet in fetched_tweets.data:
        tweet_text.append(tweet.text)
        tweet_sentiment.append(get_tweet_sentiment(tweet.text))
        tweet_id.append(tweet["id"])
        tweet_retweet_count.append(tweet["public_metrics"].get('retweet_count'))
        tweet_reply_count.append(tweet["public_metrics"].get('reply_count'))
        tweet_like_count.append(tweet["public_metrics"].get('like_count'))
        tweet_quote_count.append(tweet["public_metrics"].get('quote_count'))

    # return parsed tweets
    return pd.DataFrame({'text':tweet_text,
                         'sentiment':tweet_sentiment,
                         'id':tweet_id,
                         'retweet_count':tweet_retweet_count,
                         'reply_count':tweet_reply_count,
                         'like_count':tweet_like_count,
                         'quote_count':tweet_quote_count})
```

Figure 18. Creation of DataFrame from Figure 12



	text	id	retweet_count	reply_count	like_count	quote_count
0	Big announcement tomorrow. \nWin@Ninja Ur trash...	1538945864831488002	1153	1575	54218	121
1	Shout out to all the amazing people that helpe...	1538304556314049536	1040	749	28246	52
2	@chris He no longer looks like an alien	1538224741215084546	223	394	32774	13
3	@MorningBrew Feastables	1537904283256805731	170	378	13040	13
4	@chris I hate side quests	1537796896923721732	143	232	14086	13
...	...	...	...	...	...	...
95	@1975_mama ❤️	1519335139775557633	2	38	1097	6
96	@Koolarie No problem :)	1519335042434097159	6	39	1882	7
97	We assessed the damage after shipping out the ...	1519333780540080128	1618	3644	74279	173
98	@WalterIsaacson Can I give you everything I ow...	1518943049384136708	61	203	5570	11
99	@mikemajlak Brooooo, I forgot that was a thing...	1518789682531901440	28	86	3765	5

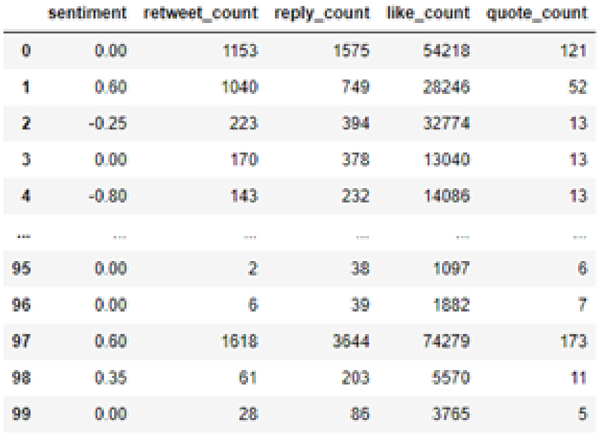
Figure 19. Starting dataset

It contains 6 attributes:

1) The text of the tweet
2) Tweet's id
3) Amount of retweets for each tweet
4) Amount of replies for each tweet
5) Amount of likes for each tweet
6) Amount of quotes for each tweet

The message text is cleaned for each tweet, and semantic analysis is performed using TextBlob.

The final dataset is shown in Figure 20.



	sentiment	retweet_count	reply_count	like_count	quote_count
0	0.00	1153	1575	54218	121
1	0.60	1040	749	28246	52
2	-0.25	223	394	32774	13
3	0.00	170	378	13040	13
4	-0.80	143	232	14086	13
...	...	...	...	...	...
95	0.00	2	38	1097	6
96	0.00	6	39	1882	7
97	0.60	1618	3644	74279	173
98	0.35	61	203	5570	11
99	0.00	28	86	3765	5

Figure 20. Final dataset

Now it contains only 5 attributes. Four of these are metrics, and the fifth is the result of the semantic analyzer. Such a dataset will be submitted for training a neural network.

Neural Network

The neural network receiving the input dataset, shown in Figure 20, is shown in Figure 21.

The input layer contains 600 neurons. It processes and receives the sentiment column as input.

Next comes 3 hidden layers with 600 neurons, a Relay activation function, and a Dropout of 0.1.

The output layer is 4 neurons since it is required to predict 4 signs. Ends with the Relu activation function.

That is, in relation to the current neural network, where Dropout is 0.1, this means that on average, 90% of all neurons will be involved because, for each neuron, the probability of disconnecting is 10%.

Figure 22 shows the creation of this neural network in python, but with the help of the Keras library, which is just created to create various neural networks.

Results and analyses

As a result of the work of the program, the following results were obtained:

For a blogger named kathryniveyy, the quote count graph is shown in Figure 23.

Like count in Figure 24.

Reply count in Figure 25.

The retweet count is in Figure 26.

As you can see in this graph, the audience of this blogger prefers an aggressive presentation of information. The number of all metrics decreases with increasing positivity. This is especially noticeable after 0.

But not all charts look the same. Figure 21 looks incredibly distinctive. But let's try to take another blogger.

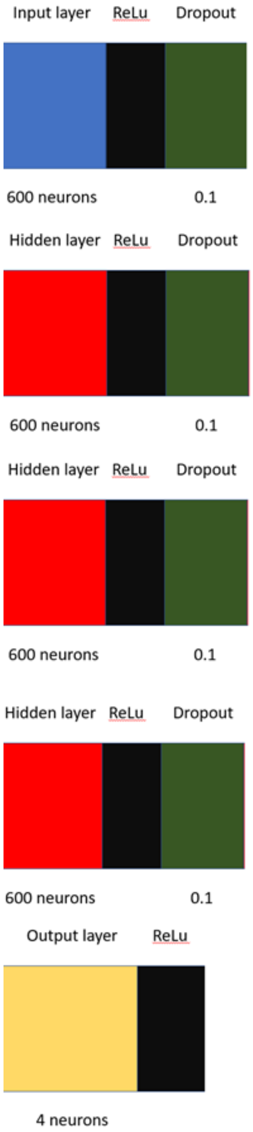
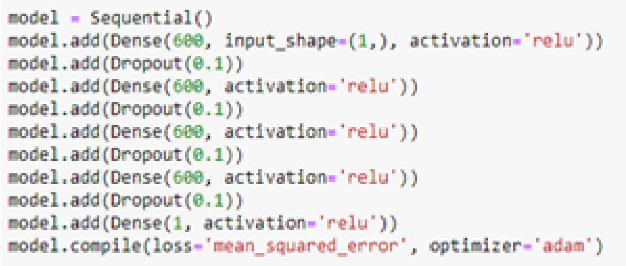


Figure 21. Neural Network scheme



```
model = Sequential()
model.add(Dense(600, input_shape=(1,), activation='relu'))
model.add(Dropout(0.1))
model.add(Dense(600, activation='relu'))
model.add(Dropout(0.1))
model.add(Dense(600, activation='relu'))
model.add(Dropout(0.1))
model.add(Dense(600, activation='relu'))
model.add(Dropout(0.1))
model.add(Dense(1, activation='relu'))
model.compile(loss='mean_squared_error', optimizer='adam')
```

Figure 22. Code representation of Neural Network

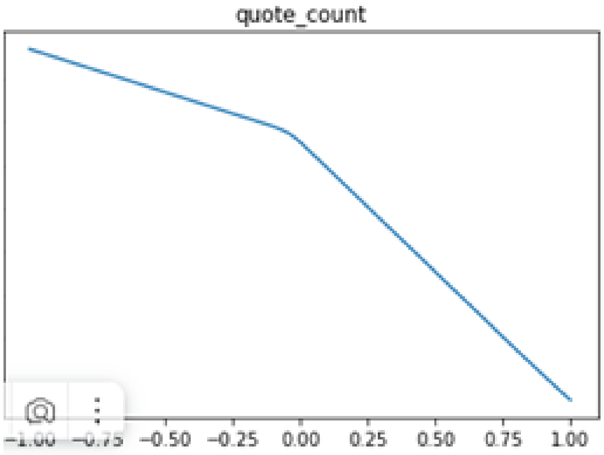


Figure 23. Quote count for kathryniveyy

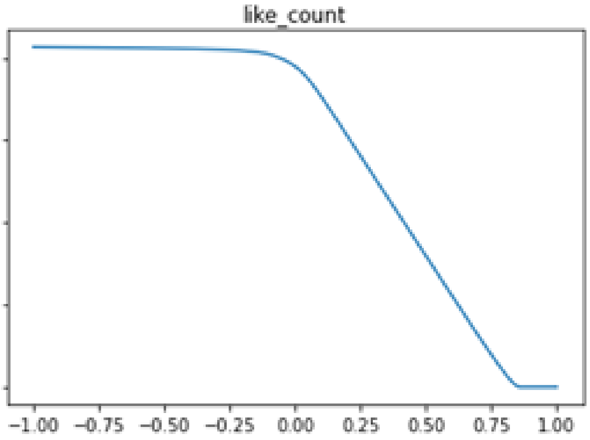


Figure 24. Like count for kathryniveyy

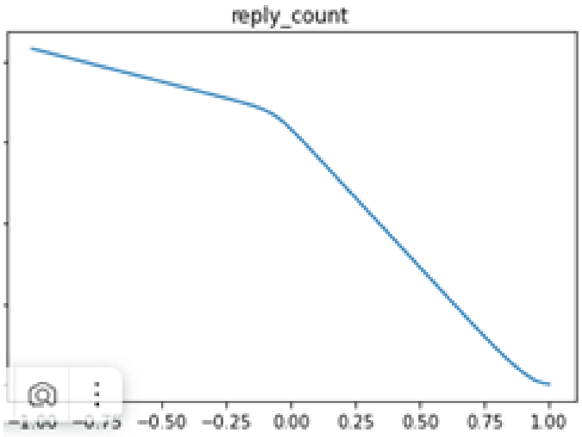


Figure 25. Reply count for kathryniveyy

For example, MrBeast.

For him, the results were the opposite.

The quote count graph is shown in Figure 27.

Like count in Figure 28.

Reply count in Figure 29.

The retweet count is in Figure 30.

As expected, all metric graphs differ from the metric graphs of the previous blogger.

If we take a prominent businessman, for example, Elon Musk, the results will be as follows:

The quote count graph is shown in Figure 31.

Like count in Figure 32.

Reply count in Figure 33.

The retweet count is in Figure 34.

As we can see, the likes chart is out of the expected picture and is quite similar to the blogger's likes chart. The rest of the metrics look about as expected.

It is also interesting to look at the results for some stores that would have an account for advertising.

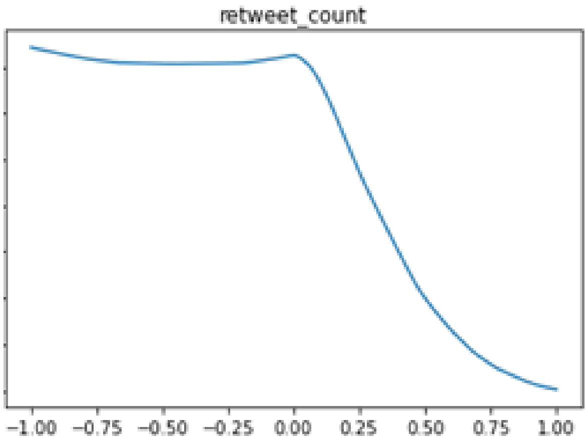


Figure 26. Retweet count for kathryniveyy

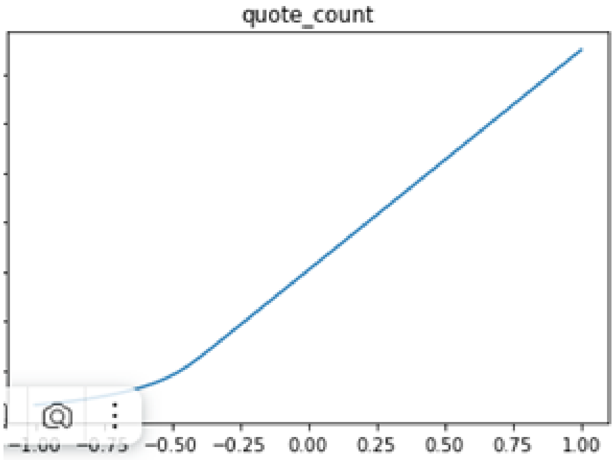


Figure 27. Quote count for MrBeast

DessertPassion was taken. For it, these results were observed:

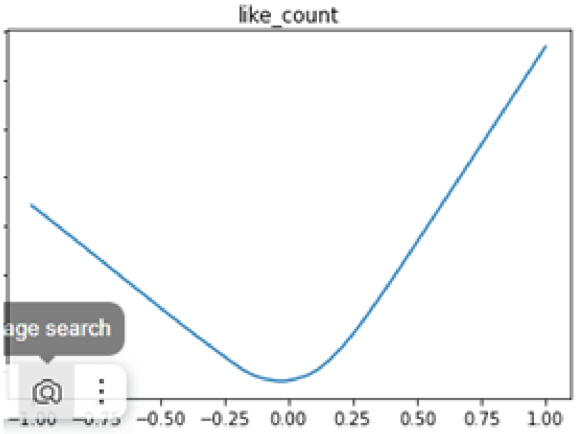


Figure 28. Like count for MrBeast

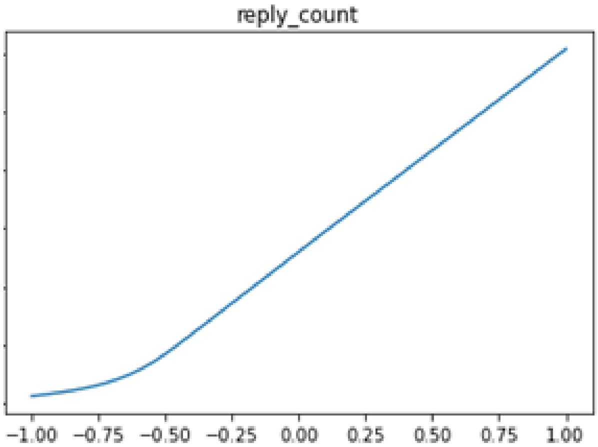


Figure 29. Reply count for MrBeast

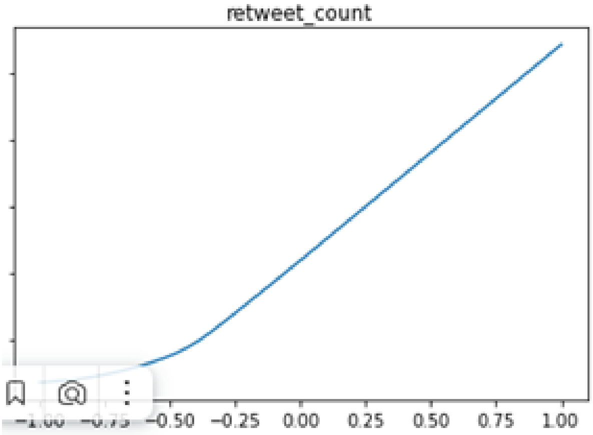


Figure 30. Retweet count for MrBeast

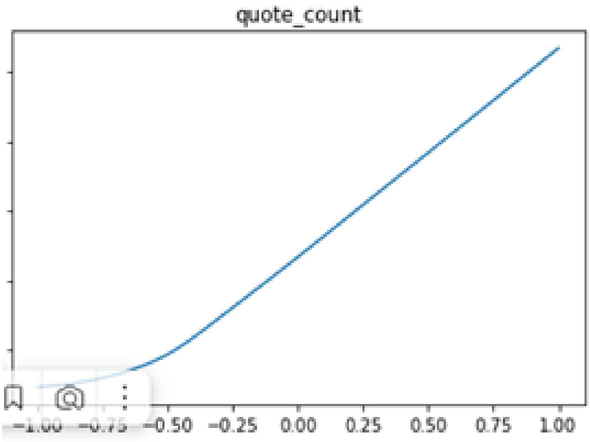


Figure 31. Quote count for Elon Musk

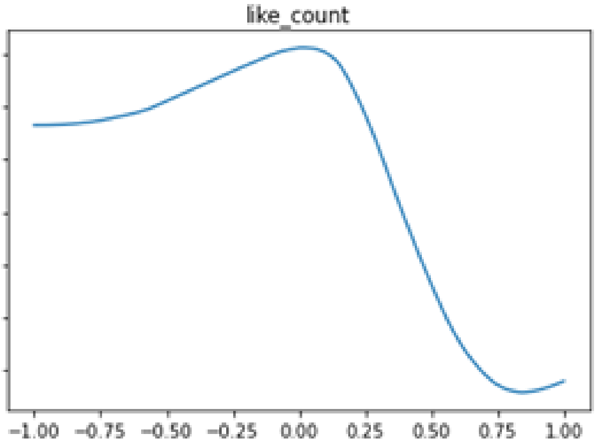


Figure 32. Like count for Elon Musk

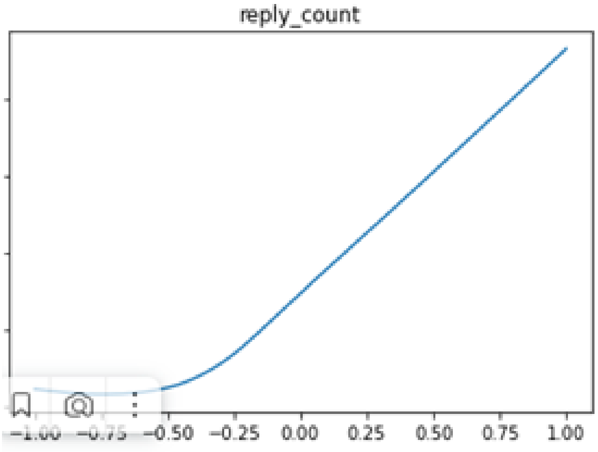


Figure 33. Reply count for Elon Musk

The quote count graph is shown in Figure 35. Like count in Figure 36.

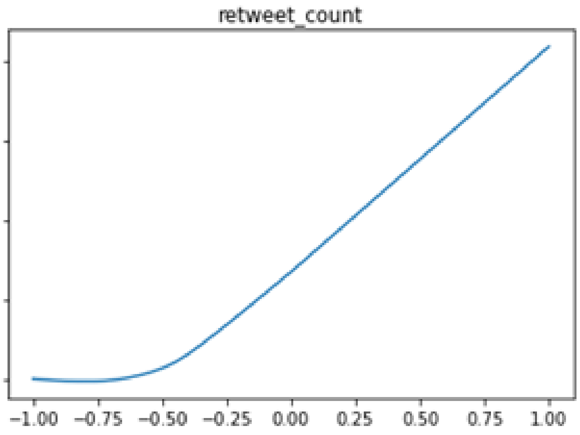


Figure 34. Retweet count for Elon Musk

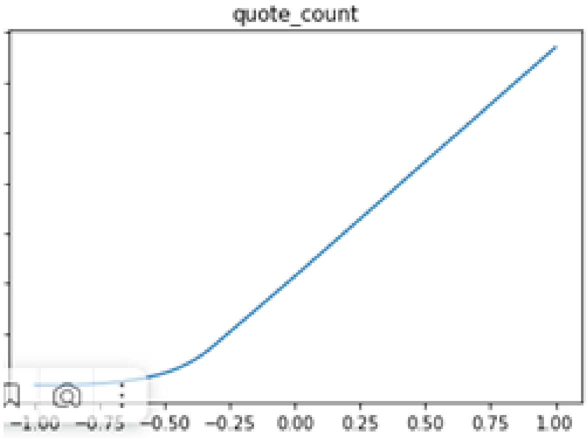


Figure 35. Quote count for DessertPassion

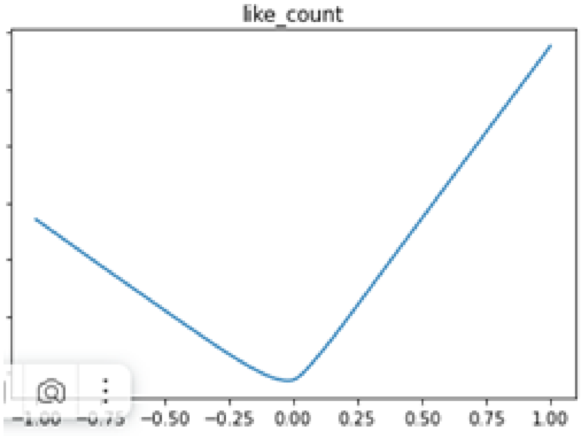


Figure 36. Like count for DessertPassion

Reply count in Figure 37.

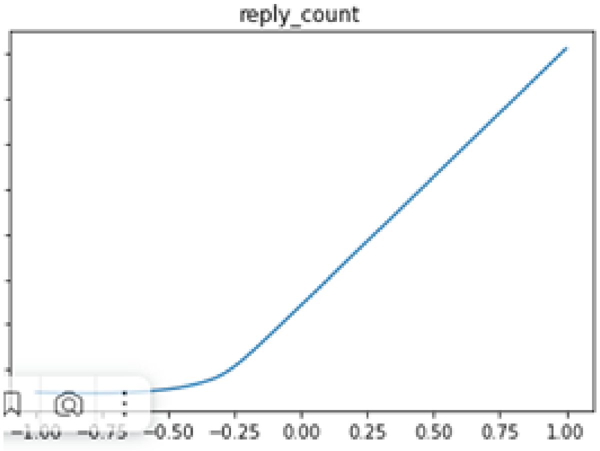


Figure 37. Reply count for DessertPassion

Retweet count in Figure 38.

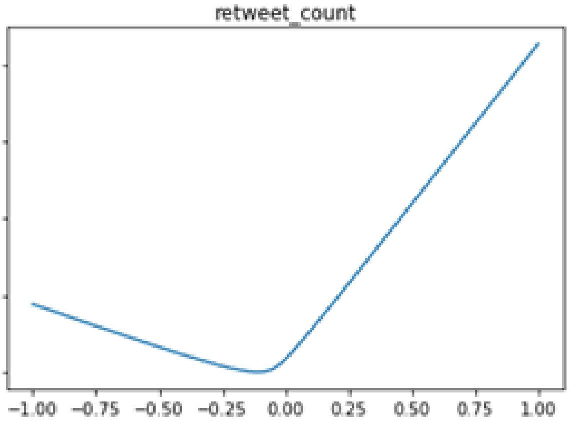


Figure 38. Retweet count for DessertPassion

The metric graphs are very similar to the metric graphs for Elon Musk. Still, since advertising implies a slightly different approach to formulating posts, we can see significant differences in the likes and small ones in the retweet graphs.

As we can see, it is impossible to choose a unique strategy for writing posts that would always work well. For different bloggers, the writing style can quickly be diametrically opposed. And even speaking of companies, the behavior of the company's face and the account for advertising will also have differences. However, they should follow the same strategy of hiding failures and demonstrating victories.

All this information is very important for those engaged in maintaining these accounts and for those who will try to analyze the behavior of people, companies, or even changes in the value of stocks.

Summary and Outlook

As a result of the work, a detailed analysis and evaluation of the capabilities of the tweety library and the situations in which it can be effectively applied was carried out, and an approximate algorithm for the operation of the semantic analyzer was described. The dataset for the model was compiled and cleaned. A neural network has also been successfully created and described, the purpose of which is to predict changes in the number of likes, reposts, quotes, and retweets from the aggressiveness of the post text for a separate profile. The results of this model were compared and described.

This project may be developed in the future to predict or analyze changes in people's positions on various issues, which will allow getting more accurate and free data than currently received by social surveys.

Another option is to supplement this algorithm to predict the trend of change in the value of stocks or even perspectives of the company.

Acknowledgments

This work was supported partially by the European Union in the framework of ERASMUS MUNDUS, Project CyberMACS (Project #101082683) (https://cybermacs.eu).

References

[1] V. Volodkin, "TWITTER History", quokka.media, https://quokka.media/istorii-brendov/twitter/ (Last access: Nov. 22, 2022)

[2] "Twitter API v2", developer.twitter.com, https://developer.twitter.com/en/docs/twitter-api (Last access: Nov. 22, 2022)

[3] "Twitter API v2 Reference", docs.tweepy.org, https://docs.tweepy.org/en/v4.4.0/client.html#tweepy.Client.hide_reply (Last access: Nov. 22, 2022)

[4] A. Kumar, "Twitter Sentiment Analysis", medium.com, https://medium.com/analytics-vidhya/twitter-sentiment-analysis-b9a12dbb2043 (Last access Nov. 22, 2022)

[5] "Neural network training", https://www.reg.ru/blog/stehnfordskij-kurs-lekciya-6-obuchenie-nejrosetej-chast-1-2/, (Last access: Nov. 22, 2022)

[6] F. Wunder, "Dropout —- a method for solving the problem of retraining in neural networks", https://habr.com/ru/company/wunderfund/blog/330814/ (Last access: Nov. 22, 2022)

[7] Mercado, Stephen C. "Sailing the Sea of OSINT in the Information Age." Secret intelligence: A reader 78 (2009).

[8] Best Jr, Richard A., and Alfred Cumming. "Open source intelligence (OSINT): issues for congress." December 5 (2007): 28.

[9] Glassman, Michael, and Min Ju Kang. "Intelligence in the internet age: The emergence and evolution

of Open Source Intelligence (OSINT).” Computers in Human Behavior 28.2 (2012): 673-682.

[10] Quick, Darren, and Kim-Kwang Raymond Choo. “Digital forensic intelligence: Data subsets and Open Source Intelligence (DFINT+ OSINT): A timely and cohesive mix.” Future Generation Computer Systems 78 (2018): 558-567.

[11] Williams, Heather J., and Ilana Blum. “Defining second generation open source intelligence (OSINT) for the defense enterprise.” RAND Corporation Santa Monica United States, 2018.

[12] Benes, Libor. “OSINT, new technologies, education: Expanding opportunities and threats. A new paradigm.” Journal of Strategic Security 6.3 (2013): 22-37.

[13] Schaurer, Florian, and Jan Störger. “The evolution of open source intelligence (OSINT).” Journal of US Intelligence Studies 19.3 (2013): 53-56.

[14] Pringle, Robert W. “The limits of OSINT: Diagnosing the Soviet media, 1985-1989.” International Journal of Intelligence and Counterintelligence 16.2 (2003): 280-289.

[15] Gibson, Helen. “Acquisition and preparation of data for OSINT investigations.” Open Source Intelligence Investigation. Springer, Cham, 2016. 69-93.

[16] Carroll, Jami M. “OSINT Analysis using Adaptive Resonance Theory for Counter-terrorism Warnings.” Artificial Intelligence and Applications. 2005.

[17] Best, Clive. “OSINT, the Internet and Privacy.” EISIC. 2012.

[18] Casanovas, Pompeu. “Cyber warfare and organized crime. A regulatory model and meta-model for open source intelligence (OSINT).” Ethics and Policies for Cyber Operations. Springer, Cham, 2017. 139-167.

[19] Layton, Robert, and Paul A. Watters. “Automating Open Source Intelligence: Algorithms for OSINT.” Syngress, 2015.

[20] Steele, Robert David. “Open Source Intelligence (OSINT).”

[21] Berghel, Hal. “Robert David Steele on OSINT.” Computer 47.7 (2014): 76-81.

[22] Weaver, Greg S. “Open Source Intelligence (OSINT).” The Police and the Military: Future Challenges and Opportunities in Public Safety 4.

[23] Revell, Quentin, Tom Smith, and Robert Stacey. “Tools for OSINT-Based Investigations.” Open Source Intelligence Investigation. Springer, Cham, 2016. 153-165.

[24] Kalpakis, George, et al. “OSINT and the Dark Web.” Open Source Intelligence Investigation. Springer, Cham, 2016. 111-132.

[25] Tabatabaei, Fahimeh, and Douglas Wells. “OSINT in the Context of Cyber-Security.” Open Source Intelligence Investigation. Springer, Cham, 2016. 213-231.

[26] Danda, Matthew. “Open Source Intelligence and Cybersecurity.” (2019).

[27] Steele, Robert D. “1997 OSINT What Is It – Why Is It Important to the Military (White Paper).” Academia.edu `www.academia.edu/9817888/1997_OSINT_What_Is_It_Why_Is_It_Important_to_the_Military_White_Paper_`, (Last access: Nov. 22, 2022).

[28] “Social Media Prisma 2017/2018”, Ethority, `ethority.de/social-media-prisma/`, (Last access: Nov. 22,2022).

[29] Mohsin, Maryam, et al. “10 Social Media Statistics You Need to Know in 2020 [Infographic].” Oberlo, Oberlo, 15 Jan. 2020, `www.oberlo.com/blog/social-media-marketing-statistics`, (Last access: Nov. 22, 2022).

[30] Tenzer: “Datenvolumen der weltweit generierten Daten 2025.” Statista, Statista, 13 Feb. 2020, `de.statista.com/statistik/daten/studie/267974/umfrage/prognose-zum-weltweit-generierten-datenvolumen/`, (Last access: Nov. 22,2022).

[31] Schwarz, Klaus; Franziska Schwarz, Reiner Creutzburg: “Conception and implementation of professional laboratory exercises in the field of open source intelligence (OSINT)”. Proceed. Electronic Imaging Symposium 2020 (San Francisco, USA), Mobile Devices and Multimedia: Technologies, Algorithms & Applications Conference (MOBMU) 2020, `https://doi.org/10.2352/ISSN.2470-1173.2020.3.MOBMU-278`, (Last access: Nov. 22, 2022).

[32] Schwarz, Klaus; Reiner Creutzburg: “Design of Professional Laboratory Exercises for Effective State-of-the-Art OSINT Investigation Tools - Part 1: RiskIQ PassiveTotal”. Proceed. Electronic Imaging Symposium 2021 (San Francisco, USA), Mobile Devices and Multimedia: Technologies, Algorithms & Applications Conference (MOBMU) 2021, `https://doi.org/10.2352/ISSN.2470-1173.2021.3.MOBMU-043`, Last access: Nov. 22, 2022).

[33] Schwarz, Klaus; Reiner Creutzburg: “Design of Professional Laboratory Exercises for Effective State-of-the-Art OSINT Investigation Tools - Part 2: Censys”. Proceed. Electronic Imaging Symposium 2021 (San Francisco, USA), Mobile Devices and Multimedia: Technologies, Algorithms & Applications Conference (MOBMU) 2021, `https://doi.org/10.2352/ISSN.2470-1173.2021.3.MOBMU-044`, Last access: Nov. 22, 2022.

[34] Schwarz, Klaus; Reiner Creutzburg: “Design of Professional Laboratory Exercises for Effective State-of-the-Art OSINT Investigation Tools - Part 3: Maltego”. Proceed. Electronic Imaging Symposium 2021 (San Francisco, USA), Mobile Devices and Multimedia: Technologies, Algorithms & Applications Conference (MOBMU) 2021, `https://doi.org/10.2352/ISSN.2470-1173.2021.3.MOBMU-045`, Last access: Nov. 22, 2022.

[35] Kant, Daniel; Reiner Creutzburg: ‘Investigation of risks for Critical Infrastructures due to the exposure of SCADA systems and industrial controls on the Internet based on the search engine Shodan”. Proceed. Electronic Imaging Symposium 2020 (San

Francisco, USA), Mobile Devices and Multimedia: Technologies, Algorithms & Applications Conference (MOBMU) 2020 `https://doi.org/10.2352/ISSN.2470-1173.2020.3.MOBMU-253`, Last access: Nov. 22, 2022.

Author Biography

Artem Sklyar received his BS in computer science from Peter the Great Saint Petersburg Polytechnic University (2021). At the same time, he worked as a Database programmer and machine learning engineer in Zolotoy 585 (Jun 2020 Mar 2021). His work has focused on the development of machine learning algorithms for retail. Now he studies Big data and AI at SRH Berlin University of Applied Sciences.

Klaus Schwarz received his B.Sc. and M.Sc. in Computer Science from Brandenburg University of Applied Sciences (Germany) in 2017 and 2020, respectively. He is currently a Ph.D. student at the University of Granada, Spain. His research interests include IoT and smart home security, OSINT, mechatronics, additive manufacturing, embedded systems, artificial intelligence, and cloud security. As a faculty member, he is developing a graduate program in Applied Mechatronic Systems focusing on Embedded Systems at SRH Berlin University of Applied Sciences.

Reiner Creutzburg is a Retired Professor for Applied Computer Science at the Technische Hochschule Brandenburg in Brandenburg, Germany. Since 2019 he has been a Professor of IT Security at the SRH Berlin University of Applied Sciences, Berlin School of Technology. He is a member of the IEEE and SPIE and chairman of the Multimedia on Mobile Device (MOBMU) Conference at the Electronic Imaging conferences since 2005. In 2019, he was elected a member of the Leibniz Society of Sciences to Berlin e.V. His research interest is focused on Cybersecurity, Digital Forensics, Open Source Intelligence (OSINT), Multimedia Signal Processing, eLearning, Parallel Memory Architectures, and Modern Digital Media and Imaging Applications.

Table 1. Tweetpy methods and their evaluation

Method	Description	Score
hide_reply()	Hides the reply to a certain tweet.	7
unhide_reply()	Unhides the reply to a certain tweet.	2
unlike()	Unlike the certain tweet.	3
get_liking_users()	Get information about users who like a tweet.	9
get_liked_tweets()	Allows you to get information about a user's liked tweets.	9
like()	Allows you to like a tweet.	1
delete_tweet()	Allows you to delete a tweet.	10
create_tweet()	Allows you to create a tweet.	0
Unretweet()	Allows you to remove the retweet of a tweet.	3
get_retweeters()	Allows you to get information about who has retweeted a certain tweet.	8
retweet()	Allows you to a tweet.	2
search_all_tweets()	Allows you to search in the history of public tweets that match a search query.	4
search_recent_tweets()	Allows you to search in the recent history of public tweets that match a search query.	5
get_users_mentions()	Allows you to get tweets that mention a single user.	0
get_users_tweets()	Allows you to get tweets composed by a single user.	10
get_all_tweets_count()	Allows you to get the complete history of public tweets that match a search query.	5
get_recent_tweets_count()	Allows you to get a complete history of public tweets for the last 7 days that match a search query.	5
get_tweet()	Allows you to get information about one tweet.	10
get_tweets()	Allows you to get information about many tweets.	10
unblock()	Allows you to unblock a user.	2
get_blocked()	Allows you to get a list of users who are blocked.	2
block()	Allows you to block a user.	2
unfollow_user()	Allows you to unfollow a user.	2
get_users_followers()	Allows you to get a list of users' followers.	2
get_users_following()	Allows to get a list of people who the user follows.	2
follow_user()	Allows you to follow a user.	2
unmute()	Allows you to unmute a user.	2
get_muted()	Allows you to get a list of users who are muted.	2
mute()	Allows you to mute a user.	2

Table 1 continuation. Tweetpy methods and their evaluation

get_user()	Allows you to get a lot of information about the user.	9
get_users()	Allows you to get a lot of information about users.	9
get_list_tweets()	Allows you to get a list of tweets from the specified List.	8
unfollow_list()	Allows you to unfollow a List.	6
get_list _followers()	Allows you to get a list of users who are followers of a List.	6
get_followed_lists()	Allows you to get a List of users to follow.	6
follow_list()	Allows you to follow a List.	6
get_list()	Allows you to get information about a specified List.	5
get_owned_lists()	Allows you to get all lists the user owns.	5
remove_list_member()	Allows you to remove a member from a List.	5
get_list_members()	Allows you to get members of the specified List.	5
get_list_memberships()	Allows you to get lists that a user is a member of.	5
add_list_member()	Allows you to add a member to a List.	5
delete_list()	Allows you to delete a List.	5
update_list()	Allows you to update the metadata of a List.	5
create_list()	Allows you to create a List.	5
unpin_list()	Allows you to unpin a List.	5
get_pinned_lists()	Allows you to get Lists pinned by a user.	5
pin_list()	Allows you to pin a List.	5

Table 2. Tweetpy methods and usage situations

hide_reply()	Creating a profile control bot can be useful. For example, if the answers are too negative, they may be hidden.
unhide_reply()	It can be useful to use to create a profile control bot. For example, if the responses are too negative and were blocked before, then after a while, they can be reopened.
unlike()	It can be useful to use it to create a profile management bot. For example, if it turns out that the one you liked loses its reputation dramatically. For example, it could be useful if you liked some posts in support of Russia before the start of the war with Ukraine, then after the start of the war, you need to delete your likes so as not to incur reputational losses.
get_liking_users()	It can be useful for analyzing your audience or people's behavior in principle. Having received information about users who have liked, you can analyze their profile and behavior on the network. For example, which people support certain ideas.
get_liked_tweets()	It can be useful for analyzing your audience or people's behavior in principle. Having received information about the tweets users have liked, it is possible to analyze their interests and views on life using semantic analysis. For example, which people support certain ideas.
like()	It can be useful for analyzing your audience or people's behavior in principle. Having received information about the tweets users have liked, it is possible to analyze their interests and views on life using semantic analysis. For example, which people support certain ideas.
delete_tweet()	It can be useful to use it to create a profile management bot. For example, to automatically delete a tweet if the reaction to it is too negative.
create_tweet()	It can be useful to use it to create a profile management bot. For example, automatically create tweets so they come out at a certain time.

Table 2 – continuation: Tweetpy methods and usage situations

Unretweet()	It can be useful to use it to create a profile management bot. For example, if it turns out that the one you retweeted loses its reputation dramatically. For example, it could be useful if you retweeted some posts in support of Russia before the start of the war with Ukraine. After the start of the war, you need to delete your retweet so as not to incur reputational losses.
get_retweeters()	It can be useful for analyzing your audience or people's behavior in principle. Having received information about users who retweeted, you can use semantic analysis to analyze their interests and even views on life. Based on this, understand which posts are liked by which categories of people.
retweet()	It can be useful to use it to create a profile management bot. For example, to automatically retweet popular tweets to generate activity in your profile.
search_all _tweets()	It can be used to create a dataset from random users, for example, using some word or hashtag for a long period of time.
search_recent _tweets()	It can be used to create a dataset from random users, for example, using some word or hashtag for a short recent period of time.
get_users _mentions()	It can be used to analyze the attitude of different categories of people to a particular user, event, or even country, as well as how this attitude changes.
get_users _tweets()	A very useful function that allows you to get user responses, and based on it, an analysis of user behavior will be performed.
get_all _tweets _count()	A very useful function that allows you to analyze the number of relevant tweets for a long period of time on request. For example, to analyze the popularity of a particular thing.
get_recent_tweets _count()	A very useful function that allows you to analyze the number of relevant tweets for a short recent period of time on request. For example, to analyze the popularity of a particular thing.
get_tweet()	Allows you to get information about a certain tweet. It can be used to analyze tweets with poor ratings.
get_tweets()	Allows you to get information about certain tweets. It can be used to analyze tweets with poor ratings.

Table 2 – continuation: Tweetpy methods and usage situations

get_all _tweets _count()	A very useful function that allows you to analyze the number of relevant tweets for a long period of time on request. For example, to analyze the popularity of a particular thing.
get_recent _tweets _count()	A very useful function that allows you to analyze the number of relevant tweets for a short recent period of time on request. For example, to analyze the popularity of a particular thing.
get_tweet()	Allows you to get information about a certain tweet. It can be used to analyze tweets with poor ratings.
get_tweets()	Allows you to get information about certain tweets. It can be used to analyze tweets with poor ratings.
unblock()	It can be useful for creating a bot. It will allow you to unblock users if you have already blocked a lot:)
get_blocked()	It can be useful for creating a bot. Allows you to find out how many and which users you have already blocked. Perhaps it's time to unblock some of them.
block()	It can be useful for creating a bot. It will allow you to block the user. It is useful if the system understands that this user is trying to compromise or promote an unfavorable position for you in the comments.
unfollow_user()	It can be useful for creating a bot. It will allow you to unfollow a user who has somehow compromised himself. It is useful not to incur unnecessary reputational losses.
get_users_followers()	It can be useful for analyzing the audience or people's behavior in principle. Getting information about the users who are following this user. With this information, you can make a complete analysis of your audience or the audience of someone else, for example, your competitor.
follow_user()	It can be useful for creating a bot. It will allow you to block the user. It is useful if the system understands that this user is trying to compromise or promote an unfavorable position for you in the comments.
unmute()	It can be useful for creating a bot. It will allow you to unmute users if you have already muted a lot:)

Table 2 – continuation. Tweetpy methods and usage situations

get_muted()	It can be useful for creating a bot. Allows you to find out how many and which users you have muted. Perhaps it's time to unmute some of them.
mute()	It can be useful for creating a bot. It will allow you to mute the user. It is useful if the system understands that this user is trying to compromise or promote an unfavorable position for you in the comments.
get_user()	It can be useful for analyzing the audience or people's behavior in principle. After receiving information about the user, you can analyze his activity, profile, and those with whom he is associated.
get_users()	It can be useful for analyzing the audience or people's behavior in principle. After receiving information about the users, you can analyze their activity, profile, and those with whom they are associated.
get_list_tweets()	It can be used to create a dataset for behavioral analysis using tweet data.
unfollow_list()	It can be used to create a dataset for behavioral analysis using data about unfollowed people.
get_list _followers()	It can be used to create a dataset for behavioral analysis using data about followers.
get_followed_lists()	It can be used to create a dataset for behavioral analysis using data about followed people.
follow_list()	It can be used to create a dataset for behavioral analysis using data about follows people.
get_list()	It can be used to create a dataset for behavioral analysis using data about unfollowing people.
get_owned_lists()	It can be used to get information about already created datasets of a certain user.
remove_list_member()	Allows you to adjust the dataset by deleting users.
get_list_members()	It will allow you to quickly get all the members of the list, which is very important for analyzing people from the list.
get_list_memberships()	It will allow you to quickly get all the members' hips of the list, which is very important for analyzing people from the list.

Table 2 – continuation. Tweetpy methods and usage situations

add_list_member()	Allows you to adjust the dataset by adding users.
delete_list()	It can be used to manage the entire dataset at once.
update_list()	It can be used to manage the entire dataset at once.
create_list()	It can be used to manage the entire dataset at once.
unpin_list()	It can be used to manage the entire dataset at once.
get_pinned_lists()	It can be used to get information about all pinned sheets at once.
pin_list()	It can be used to manage the entire dataset at once.