big data and cognitive computing

MDPI

Article

Twitter Analyzer—How to Use Semantic Analysis to Retrieve an Atmospheric Image around Political Topics in Twitter

Stefan Spettel [1] and Dimitrios Vagianos [2,*]

[1] Faculty of Informatics, Technical University of Vienna, Karlsplatz 13, 1040 Wien, Austria
[2] Department of International and European Studies, University of Macedonia, Egnatia 156, 54636 Thessaloniki, Greece
* Correspondence: vagianos@uom.gr; Tel.: +30-2310-891-487; Fax: +30-2310-891-285

Received: 5 June 2019; Accepted: 2 July 2019; Published: 6 July 2019

Abstract: Social media are heavily used to shape political discussions. Thus, it is valuable for corporations and political parties to be able to analyze the content of those discussions. This is exemplified by the work of Cambridge Analytica, in support of the 2016 presidential campaign of Donald Trump. One of the most straightforward metrics is the sentiment of a message, whether it is considered as positive or negative. There are many commercial and/or closed-source tools available which make it possible to analyze social media data, including sentiment analysis (SA). However, to our knowledge, not many publicly available tools have been developed that allow for analyzing social media data and help researchers around the world to enter this quickly expanding field of study. In this paper, we provide a thorough description of implementing a tool that can be used for performing sentiment analysis on tweets. In an effort to underline the necessity for open tools and additional monitoring on the Twittersphere, we propose an implementation model based exclusively on publicly available open-source software. The resulting tool is capable of downloading Tweets in real-time based on hashtags or account names and stores the sentiment for replies to specific tweets. It is therefore capable of measuring the average reaction to one tweet by a person or a hashtag, which can be represented with graphs. Finally, we tested our open-source tool within a case study based on a data set of Twitter accounts and hashtags referring to the Syrian war, covering a short time window of one week in the spring of 2018. The results show that while high accuracy of commercial or other complicated tools may not be achieved, our proposed open source tool makes it possible to get a good overview of the overall replies to specific tweets, as well as a practical perception of tweets, related to specific hashtags, identifying them as positive or negative.

Keywords: sentiment analysis; semantic analysis; Social Media analysis; Twitter analysis

1. Introduction

Since the introduction of social media platforms, their impact has gradually increased. Many people use social media sites like Facebook and Twitter as a primary source of information. People exchange personal data and give their friends and followers the chance to stay updated on what is going on in their lives. On the other hand, online contemporary societies tend to polarize into subgroups of individuals with dramatically opposite perspectives [1,2]. This phenomenon is reflected—and often amplified—in online social networks, where humans have recently ceased to be the only players, now coexisting alongside bots (software control accounts) [3]. Consequently, social media platforms can be envisaged as rich sources of data for opinion mining and sentiment analysis [4]. The Cambridge Analytica scandal in 2018 showed that there are companies specialized in extracting, categorizing,

and using data retrieved from social networks for political purposes. In 2018, it appeared that about 50 million Facebook profiles had been analyzed by a company for political purposes within the US Election framework [5]. Although the company claimed the success of their method, called micro-targeting, the real implications on the outcome of the US election are highly debatable. It has been shown that available data has already been used to try to retrieve an exact view of the opinions of people about certain political topics.

Due to their particular features, microblogging platforms provide the scientific community with a vast variety of data that can be investigated using several mining techniques. Twitter, as a popular representative of that kind of site, is also heavily used to express political opinions, and to stay informed about political topics. The way people are becoming informed about political topics online has a measurable impact on the offline world. The way political parties, people, and institutions express their opinions can, for example, have a huge impact on the way voters decide during elections. This trend has also been recognized by politicians and decision-makers, leading to a statement from the former EU commissioner Günther Öttinger, as well as the German chancellor Angela Merkel, who described digital personal data as the "resource of the future" [6]. The latest developments around the Brexit referendum and especially the election of US President Trump have shown that politicians and political parties are willing and already using this resource to influence elections in their favor. Ahmed and colleagues performed a comparison of campaign strategies involving Twitter in the 2014 Indian elections, which was the country's first experiment with using social media for political campaigning [7].

There have been very helpful approaches in the past that aimed in detecting political trends on twitter. For instance, Stella and colleagues [1] recently used sentiment analysis in multiple languages for reconstructing and identifying political movements and political atmospheric images in the Catalan referendum. For the same event, Vinayakumar and colleagues performed stance and gender detection according to the user tweets [8]. Furthermore, Ozvuk and Ayvaz [9] used sentiment analysis for understanding the general perception of the English speaking and Turkish speaking communities towards the human emergency of Syrian refugees. Both these approaches outline the social importance of being able to parse sentiment polarities and trends in social media for understanding voting or humanitarian events.

There have also been scientific attempts that examined the influence of Twitter in journalism. Lee and colleagues examined factors that affect journalists' Twitter use behaviors in South Korea [10], where Wang and colleagues developed a system designed for real-time analysis of public sentiment toward presidential candidates in the 2012 US election, as expressed on Twitter.

This paper consists of the following parts:

- Related past approaches to this field of study are highlighted, while the contributions of TwitterAnalyzer to the scientific community is explained;
- Analysis of the applied methodology;
- A thorough description of the implementation of the TwitterAnalyzer tool;
- Testing the TwitterAnalyzer over tweets related to the Syrian civil war and the entailing findings;
- Discussion of the results of the test;
- Conclusions deriving from the testing procedure in accordance with the principles of the applied methodology and the steps taken for the tool's implementation, along with suggestions for further improvements and work.

2. Related Work and Contributions

There are currently many Twitter analysis tools available for the general public. These tools can be used to get detailed statistics about the outreach and popularity (by views, shares, etc.) of a user's own content. Nevertheless, it is not easy to find a publicly-available sentiment analysis tool that can be used to investigate the opinions and meanings of online tweets concerning specific topics. Therefore,

it is important to create open-source tools which can be used with no restrictions by researchers to retrieve statistics about the public opinion in a variety of fields. Open-source tools are set to the disposal of the researchers' community through public repositories for further additions and constant improvement. This can contribute to an increasing interest in this field of research of great importance. Public opinion can be analyzed through the public discussion, as it is reproduced by peoples' tweets around those topics.

There have been some important approaches in the past that used open techniques for sentiment analysis in tweets. For instance, the extensive work by Mohammad and colleagues on detecting positive and negative emotions from hashtags [11] was based on a state-of-the-art sentiment analysis system of short informal textual messages, such as tweets and Short Message Service (SMS). Their system was based on a supervised statistical text classification approach by leveraging a variety of surface-form, semantic, and sentiment features. The sentiment features were primarily derived from novel high-coverage tweet-specific sentiment lexicons that were automatically generated from tweets with sentiment-word hashtags and from tweets with emoticons. The same group recently released a package named AffectiveTweets, for detecting sentiment and emotional patterns in Tweets [12]. It has been implemented as a package for the Weka machine learning workbench and provides methods for calculating state-of-the-art affect analysis features from tweets, that can be fed into machine learning algorithms implemented in Weka. It also implements methods for building affective lexicons and distant supervision methods for training affective models from unlabeled tweets.

Another relevant tool is Sentiment Viz, developed by NC State University, which features an online graphical user interface (GUI) for visualizing sentiment patterns of tweets with a given keyword—word clouds and sentiment in plane style graphs are some of its features. Hutto and Gilberg [13] also presented a simple rule-based model for general sentiment analysis of general social media text. They used a combination of qualitative and quantitative methods and first constructed and empirically validated a gold-standard list of lexical features, which were specifically attuned to sentiment in microblog-like contexts. They then combined these lexical features with consideration for five general rules, which embody grammatical and syntactical conventions for expressing and emphasizing sentiment intensity. They created a tool called VADER (Valence Aware Dictionary and sEntiment Reasoner) based on those methods.

All the above-mentioned approaches combine complicated qualitative and quantitative techniques that, in most cases, require the possession of programming skills, artificial intelligence, and machine learning experience [14], along with a deep knowledge of handling and measuring big amounts of data. This combination, although proven impressive in many cases (VADER's creators proved an F1 classification accuracy of 0.96), fails to attract, and sometimes discourages, new researchers in the field of sentiment analysis over short texts produced by social media and consequently microblogging platforms' accounts.

This paper describes the implementation of a comparatively simpler tool, here called "TwitterAnalyzer" using open-source side components. Its usefulness relies on the fact that it can provide simple sentiment classification, and its simplicity and open collaboration principles can be further developed by the researcher community, aiming to meet the progressively higher standards of the approaches described above. Additionally, as will be presented in the technical implementation section, the modularity of TwitterAnalyzer allows for an easy exchange of the sentiment analysis libraries that it exploits in order to test which one provides the best results.

For testing and demonstrations purposes it was used over Twitter data related to the Syrian War in the spring of 2018 for a limited time window of one week, in order to perform top level opinion mining and sentiment analysis.

3. Methodology

To produce an atmospheric image of specific topics on Twitter and perform effective opinion mining, several steps were necessary. The first step was, unavoidably, to collect all the related tweets.

According to Saura's Three-Stage Methodological Process of Data Text Mining [14], sentiment analysis constitutes the second stage. In accordance with this approach, this paper focused on this stage and aimed to present a tool that will serve this intermediate stage of all similar analysis approaches.

In order to identify the related tweets, hashtags were exploited. It is possible to encounter Twitter accounts which produce messages mainly about one topic, for example journalists, foreign correspondents, governmental bodies, embassies, and many more. To collect those tweets automatically, a tool needs to connect to Twitter in order to retrieve the data. There are two main possibilities of how the data can be retrieved: via the search API and via the stream API. API stands for application programming interface and can be used by anybody that is using Twitter to retrieve data. The search API can deliver historic data, for example, tweets which were posted about a week ago. The stream API needs to actively listen for new events and therefore the tool needs to be running continuously.

There are many advantages and disadvantages of both techniques. For this study, the streaming API was used, mainly for the reason that the search API does not guarantee completeness and therefore it was not possible to retrieve enough data for this specific case study. Rate limiting is different as well. Rate limiting is used by Twitter to only allow a certain amount of connection attempts to specific resources per time frame. Those rate limits do not apply for premium APIs, where business partners can pay for prioritized access to the data. As this project aimed to have a tool for individuals and smaller groups, the premium API could not be used here. The public API's rate limit allows 450 search requests per 15 min time-window. Therefore, in one day, about 43,200 search requests can be performed (24 h $\times$ 4 windows/hour $\times$ 450 requests/window), whereas each request can result in up to 100 tweets [15]. For this application, it might happen that search requests deliver empty results; therefore, it is not realistic to assume that one can retrieve about 4 million tweets every day. More details about the methods for retrieving tweets can be found in Section 4. The rate limit using the streaming API is not that well-defined, although it guarantees complete delivery of tweets, unless the tweets to be received return more than one percent of the whole tweets tweeted on Twitter at that moment. Considering that about 6000 [16] tweets are posted each second, one stream can receive a maximum amount of about 60 streams per second, which results in about five million Tweets per day, considering a scenario where the number of tweets for a topic do not vary over the day.

Once the tweets were received, the tool needed to process the tweets one by one and save the results in a database. Using stream API, the sequence was as follows: the moment a tweet was sent, the program received a copy of the tweet object, when this object was either somehow connected to one of the persons to follow or contains one of the hashtags of interest. In cases where the tweet was a new tweet by a person of interest, this tweet was saved in the database. In cases where the tweet was sent by somebody or was a response to a tweet of somebody belonging to the list of persons, the sentiment was retrieved and the response stored in the database together with the sentiment. In cases where the tweet or the reply it was referring to were not yet in the database, they was added. If a tweet was a direct tweet (not a response) to one of the persons to follow, it was not evaluated. If the tweet contained a hashtag of interest, the sentiment and tweet text were retrieved as well and stored in the database.

As written above, the stream API delivers tweets in real time. This, of course, has the disadvantage that when using it, somebody cannot go back in time and the tool will only receive data from time t = 1, where the tool has been set to follow a person or a hashtag. This fact annotates a big difference in comparison with a tool like Sentiment Viz—the scope of TwitterAnalyzer is to follow a Twitter account or hashtag and get the aggregated sentiment of all the replies to tweets issued by that account. That could be very useful if somebody wants to track how positively or negatively a tweet has been received by the people who replied. This can lead to a huge amount of data (e.g., in average 1500 replies to only one tweet of Donald Trump, resulting in approximately 120,000 total replies compared to 320 results in sentiment viz when searching for @realdonaldtrump). This approach can lead to long-term studies, as all the data are collected as long as the tool is running.

The sentiment analysis itself is the most important but also most complicated and error-prone part. It is the most important core piece of tweet sentiment analysis and it is also the part where it

gets very difficult to achieve high success rates. Currently, this project used an open-available Python library called TextBlob [17], which is based on NLTK (Natural Language Toolkit) [18] and Pattern [19]. There are currently two major different methods available in TextBlob for classifying a given text. The pattern-based method uses pre-defined data of classified adjectives to describe the polarity, which is the sentiment or indication whether a statement is positive or negative. "Written text can be broadly categorized into two types: facts and opinions. Opinions carry people's sentiments, appraisals and feelings towards the world. The pattern.en module bundles a lexicon of adjectives (e.g., good, bad, amazing, irritating) that occur frequently in product reviews, annotated with scores for sentiment polarity (positive ↔ negative) and subjectivity (objective ↔ subjective)."

This library for sentiment analysis has about 75% accuracy [19]. The other approach uses a Naive Bayes Analyzer, which works with probabilities and describes the probability that a given tweet is positive or negative. To accomplish this, the Analyzer needs to be given a set of training data, which are already classified into sentences and texts. The machine-learning algorithm can then decide based on the learned model if a new, unknown text can be classified as positive or negative. NLTK provides a pre-learned model with 2000 movie reviews.

When it comes to the application of the two above mentioned libraries, there are some aspects about the communication in social media which need to be considered. First of all, predefined training data and tests about accuracy are mostly done with movie reviews. Unfortunately, there is a difference between the way people write movie reviews and a typical political tweet. Movie reviews are subjective, but the best reviewers try to give good explanations about the decisions they made for the rating, as well trying to be as objective as possible. In Twitter, similar to normal communication outside of the digital world, the language is more subtle. Sarcasm and irony are very commonly used forms of expression. Obviously, it is nearly impossible for a computerized sentiment analysis to clearly recognize sarcasm, as it is necessary to see it in the embedded context.

In this project, the pattern analyzer was used due to its simplicity. Nevertheless, if a framework for publicly-available sentiment analysis should be created, the accuracy of the results should be higher than 82%, which is the human accuracy score in classifying sentiment polarity. This can be achieved with a machine-learning approach. If this technique is used, it is very important that there is adequate available training data. This data needs to be created—at least to a certain degree—manually, by humans, which is very time-consuming. In the future, it will be necessary to create good publicly-available training data to improve the accuracy of the currently available tools. Most probably, companies like Cambridge Analytica, with sufficient human and financial resources available, are already in possession of internal, tailored training data to improve the accuracy of the analysis of Tweets and Facebook posts for political topics.

4. Technical Implementation

TwitterAnalyzer was created using Python 2.7 and used the Python wrapper tweepy to access the Twitter API. For the storage of the tweets and the sentiment results, a MySQL database was implemented and the python package textblob was used for sentiment analysis. The source code is available at https://gitlab.com/spuddl/TwitterAnalyzer.

4.1. Twitter API

The Twitter API [20] gives programmers a wide range of possibilities. It enables very easy access to Twitter data objects. Those objects are represented in JavaScript Object Notation (JSON), a file format used for communication in the web. The objects contain all the information which is necessary to interact with anybody. Tweet objects contain the information of a tweet, which is represented by a unique ID. It contains the date of creation, the content, the unique ID, and a reference to the user which created the tweet. Apart from that, there is much more information available, like geo-location, the ID of the tweet which had been replied to, and many more. The user objects store all relevant and publicly available information of a user, including the date of creation, unique user ID, name, location,

and several descriptions, including profile pictures. Having the user's ID, it is possible to retrieve their timeline or get a list of all the followers or all the people they are following. To have access to those features, a Twitter app needs to be created to generate access keys. It can be registered under https://apps.twitter.com/. It is mandatory to provide a mobile phone number for the registration of a Twitter app.

4.2. Python Modules

The following Python modules were used for implementation of TwitterAnalyzer.

1. Tweepy—Python Wrapper for the Twitter API [21].
2. TextBlob—Python Library for processing textual data [17].
3. MySQL-python—Python Wrapper for the access to a MySQL database [22].
4. matplotlib—Python 2D plotting library for graphs [23].

4.3. Modules and Structure

The class diagram of Figure 1 describes the classes and scripts (static classes) used in the project. The *start_analyser* was the entry point for the code execution. It read the configuration from the configuration file, authenticated the application to Twitter, created the database access object, and finally called the *stream_handler* to listen to incoming tweets.

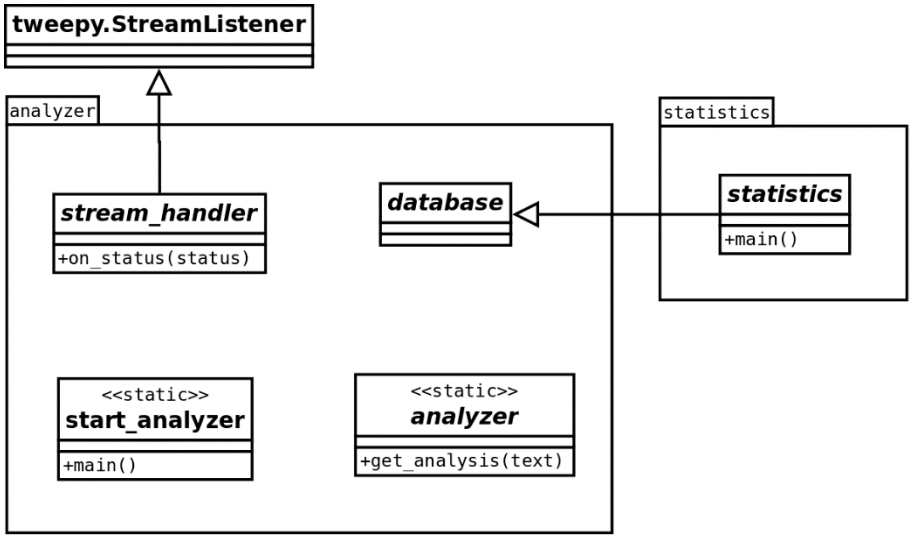


Figure 1. Static classes and scripts used in the project.

The *analyzer* class used the pattern-based sentiment analysis provided by the TextBlob Python package. It was the *get_analysis* method that returned the sentiment based on the text provided.

The database class was responsible for handling all connections to the MySQL database (Figure 2). Its methods performed the SQL statements to insert, update, or select data in/from the database and return python objects. Therefore, no other class (except statistics) needed to execute SQL statements. The database class provided methods to add a user, add a hashtag, retrieve hashtags and users, as well as adding new tweets and new replies to those tweets which contained the sentiment.

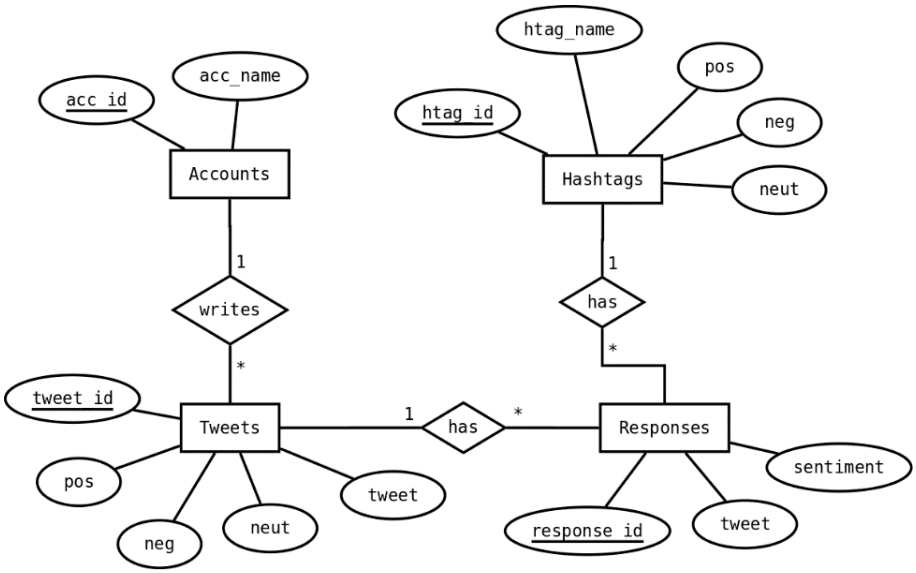


Figure 2. Database design.

The *stream_handler* was fed by the class *StreamListener* from the tweepy package. It provided a convenient way to access incoming tweets, which were received based on the lists of persons and hashtags the user defined. Whenever a tweet was received, the *on_status()* method was executed. First, it checked whether the received tweet contained any hashtags of the hashtag list, then it checked if there was a tweet by any person in the person list, and finally it checked if it was a reply to any of the tweets. In cases where it was a reply to anybody in the person list, but the tweet was not available in the database yet, this specific tweet (which was indicated by the ID in the reply tweet) was downloaded and saved in the database. In cases where none of the scenarios described above occurred, the tweet was ignored. To check if a tweet was a reply or not, it was not very straightforward, as the Twitter API did not give a list of replies for each tweet. The status object contained the name of the account to which the tweet was replying. With this name, all the past tweets from this person could be found in the database. If one was not the author of the reply but one of the persons in the person list, the reply tweet then had a parameter for the tweet ID of the original tweet it had replied to. If this ID was the same as any of the IDs of past tweets, it was a reply to one of the tweets in the database. In that case, the sentiment was calculated and the values updated to the database. If this ID was not in the database, the original tweet was downloaded.

It is important to mention that the Python program analyzer needed to run continuously, as the stream pattern was not allowed to access past data. Therefore, in cases where the program was not running and listening to new tweets, those tweets were not taken into account in the analysis.

The *statistics* script was used to access all the data in the database and save bar charts and figures. It was able to plot bar charts with total values per account and average values per account, as well as plot charts for every day. Obviously, this script needed to be executed after the analyzer had collected enough data.

4.4. Installation and Deployment

In order to execute the application, several Python dependencies, as well as the MySQL Server, needed to be installed. Detailed installation instructions can be found on GitLab: https://gitlab.com/spuddl/TwitterAnalyzer.

As outlined in Section 4.3, the analyzer tool needs to run continuously. Therefore, the tool should be deployed on a server with a steady Internet connection and enough space for the growing database.

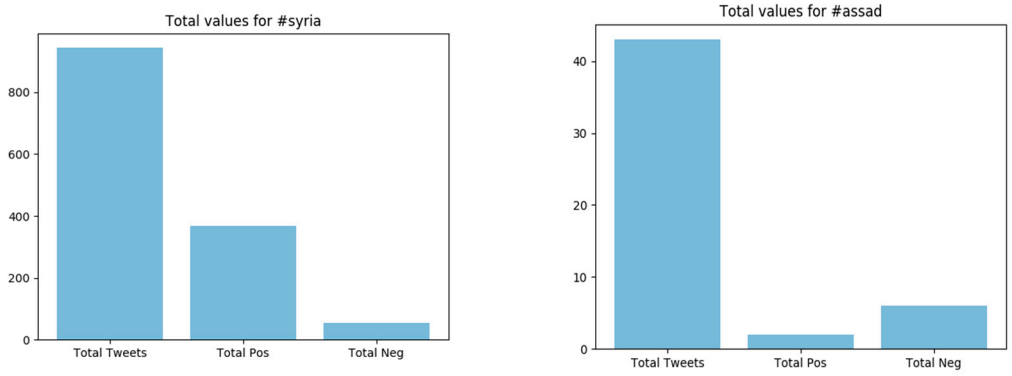
5. Testing with Tweets Related to the Syrian Civil War and Findings

To test the created software, it was necessary to let the system collect data about specific political topics. Therefore, the tool was configured to follow several accounts related to Syria, belonging to officials, journalists, and organizations, as well as following hashtags concerning Syria.

This topic was chosen as in May 2018, the situation in Syria seemed to get worse after a chemical gas attack on the city of Douma on 7 April [24]. This incident led to missile strikes against multiple government sites in Syria by the United States, France, and the United Kingdom [25]. The analyzer ran for five consecutive days, from 21 May to 25 May. In this short period of time, about 1100 tweets were collected, about 950 of which were marked with the hashtag "#syria". The others consisted of replies to tweets from political bloggers or journalists. This is, of course, not a huge amount of data, given the fact that only Donald Trump, who was also in the list for testing purposes, had generated about 135,000 responses in those days. The people who were followed were realdonaldtrump, BowenBBC, samdagher, USEmbassySyria, LizSly, cjchivers, BBCLinaSinjab, DavidKenner, NoahShachtman, Raniaab, hxhassan, MahirZeynalov, fpleitgenCNN, and SyriaCivilDef. Apart from these, the hashtags syria, syriacrisis, and assad were tracked.

The study has shown that the classification of the data does not work perfectly. There were some false positives and some false negatives which need to be corrected manually. The tweet to Donald Trump, "@realDonaldTrump Do those so called heroes know that you created, funded and trained the terror groups such as Taliban, Al Qaeda and most recently ISIS? #OsamaBinTrump", was classified as positive, whereas it is obvious that this was a very negative comment. Other comments were also difficult to classify, for example, "Love you President Trump!! You are the only president in history to follow through with so many promises. I know you don't listen to the blind, deaf and dumb sheep in here. They're just mad they've been deceived by their masters." The first part of the statement is very positive and refers to the person Donald Trump, the rest is very negative and refers to "the others". Nevertheless, this comment was classified as negative by the applied sentiment analysis tool.

The statistics tool created many interesting charts based on the data which were analyzed. For example, it can be seen that most of the Tweets containing "#syria" were classified as positive, and (English speaking) tweets with "#Assad" were mostly classified as negative, although the sample size in the last case was too small to predict a reliable trend. The numbers in the charts of Figure 3 describe the total number of tweets containing the hashtags, and the ones of these which were classified positive or negative within those five days.

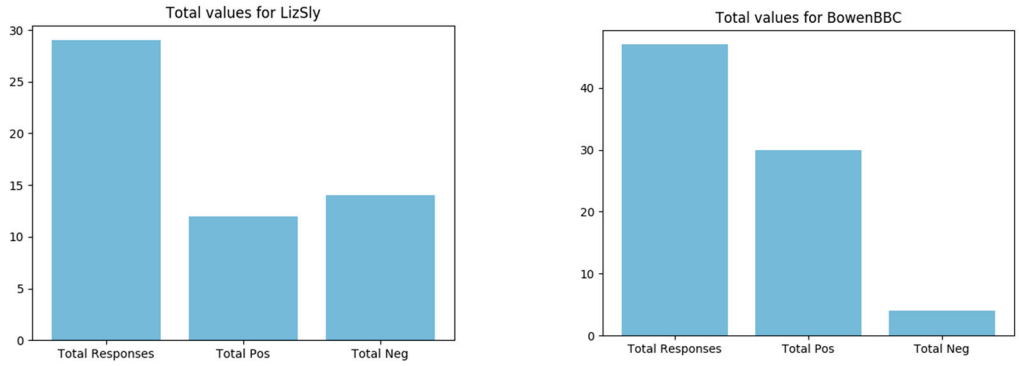


(a) (b)

Figure 3. Total values for: (**a**) #syria; (**b**) #assad.

The BBC Middle-East journalist Jeremy Bowen (*@BowenBBC*) got mainly positive replies to his own tweets (Figure 4). The Washington Post bureau chief "Liz Sly" in Beirut, covering Syria, Lebanon, Iraq, and beyond, received more negative replies, although the total numbers were so small that it was

impossible to get a reliable picture, especially in consideration of the small sentiment analysis success rate of only 75%.



(a) (b)

Figure 4. Total values for: (**a**) LizSly; (**b**) Jeremy Bowen.

6. Discussion

As described in Section 3, the biggest challenge in completing this task was to increase the reliability of the sentiment analysis. The handling of the database, as well as the downloading of the content of Twitter, is quite straightforward, but there is no easy way to enhance the accuracy of the sentiment analysis. An accuracy of 75% might not be enough for most applications, although it might be enough to show a trend. Nevertheless, those numbers and statistics should always be handled carefully.

In an attempt to compare TextBlob with pre-existing libraries like VADER, it could be stated that its main advantage would be the fact it can be automatically trained on a given corpus, thus not requiring the lexicon polishing that VADER does. In any case, VADER embodies more sophisticated mechanisms than TwitterAnalyzer which, as stated earlier, was developed in order to demonstrate the procedure of implementing a simple tool that can be set to the disposal of multivariate skill and knowledge levels researcher, who can use it for getting initial results, exchange components (e.g., libraries) to perform multiple tests, or further improve it according to their personal scientific background. During this procedure, quantitative data can be acquired in an attempt to measure indices like classification accuracy, and further evaluate the overall performance of the modified versions of the tool. These measurements have not been made for TweeterAnalyzer, as this procedure was not within the scope of this paper.

It would also be interesting for many research projects to access the semantic content of positive and negative tweets. A relatively simple-to-do visualization is always a word cloud [9], highlighting the most frequent concepts or words in positive and negative tweets.

Finally, it should be underlined that stance [8] is different from sentiment. This is an important point that has been cited above by giving some specific examples of tweets. Researchers should always bear in mind that using only sentiment analysis only could be problematic, because no matter what the monitoring tool might be, a short text cannot include enough information for automatic tools to automatically derive stances or attitudes from it.

7. Conclusions and Further Work

As mentioned earlier, due to its open architecture, this analyzer class can further be developed and enhanced by adding more features. The sentiment analysis is the core piece of this application and is currently very basic, as it heavily relies on the accuracy of the TextBlob provided methods. It would also be possible to perform the sentiment analysis not only on the whole tweet, but also on all the sentences of each tweet, or even on parts of each sentence. Sooner or later, researchers will further

develop a tool like this, shaping it to follow their standards, and it will be necessary to use another approach and use classifiers, like the Naive Bayes Classifier, which have initial training data, but also give the users the possibility to train the algorithm over time. A good extension would be to provide the user or administrator with a list of a randomly chosen sample (e.g., 100 replies) out of the database. Those samples contain the text as well as the algorithm-determined sentiment. The user is then able to correct the sentiments. Those corrections will be saved in the database. It can also serve as additional and verified training data for the algorithm. After the "training" of the algorithm, the whole database (or just a subset) could be updated. This could ensure a higher accuracy, although it might also be necessary to pre-select tweets for the training data, as the quality of the training data should be high.

Future research related to this work could focus on collecting data over a longer time period, for example, over several weeks, and provide a detailed analysis on how accurate the current semantic function is by taking samples from the categorized data out of the database. Overall, the size of the sample area and the semantic area field of any similar application are highlighted as key factors that determine the success of using similar approaches in this field of study [14]. Apart from that, the algorithms used for semantic analysis can be exchanged or improved to increase the accuracy. From a global perspective, the need for developing sentiment analysis tools and training algorithms over multilingual corpus will soon emerge. The comparison of the results in several languages will point out new interesting aspects of multilingual sentiment classifiers' applications [4].

In the present-day digital ecosystem, where more data are generated daily and influence the public opinion, sometimes by fake accounts (bots) [26], new methodological contributions for automatic or semi-automatic effective analysis will be highly appreciated [14].

Author Contributions: Conceptualization, S.S. and D.V.; Methodology, S.S. and D.V.; software, S.S.; writing—original draft preparation, S.S.; writing—review and editing, D.V.

Funding: This research received no external funding.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Stella, M.; Ferrara, E.; De Domenico, M. Bots increase exposure to negative and inflammatory content in online social systems. *Proc. Natl. Acad. Sci. USA* **2018**, *115*, 12435–12440. [CrossRef] [PubMed]
2. Conover, M.; Ratkiewicz, J.; Francisco, M.; Goncalves, B.; Menczer, F.; Flammini, A. Political polarization on Twitter. In Proceedings of the International Conference on Weblogs and Social Media, Palo Alto, CA, USA, 17–21 July 2011; pp. 89–96.
3. Bessi, A.; Ferrara, E. Social bots distort the 2016 US presidential election online discussion. *First Monday* **2016**, *21*. [CrossRef]
4. Pak, A.; Paroubek, P. Twitter as a corpus for sentiment analysis and opinion mining. In Proceedings of the Seventh Conference on International Language Resources and Evaluation (LREC'10), Valletta, Malta, 19–21 May 2010; Volume 10, pp. 1320–1326.
5. Tilman Wittenhorst. Trump-Wahlhelfer Cambridge Analytica: Streit um 50 Millionen Facebook-Profile. 2018. Available online: https://www.heise.de/newsticker/meldung/Trump-Wahlhelfer-Cambridge-Analytica-Streit-um-50-Millionen-Facebook-Profile-3997820.html (accessed on 22 June 2018).
6. Albert Funk, Merkel: Daten Sind der Rohstoff der Zukunft. 2015. Available online: https://www.tagesspiegel.de/wirtschaft/digitalisierung-der-wirtschaft-merkel-daten-sind-der-rohstoff-der-zukunft/12312978.html (accessed on 22 June 2018).
7. Ahmed, S.; Jaidka, K.; Cho, J. The 2014 Indian elections on Twitter: a comparison of campaign strategies of political parties. *Telemat. Inform.* **2016**, *33*, 1071–1087. [CrossRef]
8. Taule', M.; Martí, M.A.; Rangel, F.M.; Rosso, P.; Bosco, C.; Patti, V. Overview of the task on stance and gender detection in tweets on Catalan independence at IberEval 2017. In Proceedings of the 2nd Workshop on Evaluation of Human Language Technologies for Iberian Languages, IberEval 2017, Murcia, Spain, 19 September 2017; Volume 1881, pp. 157–177.

9. Öztürk, N.; Ayvazb, S. Sentiment analysis on Twitter: A text mining approach to the Syrian refugee crisis. *Telemat. Inform.* **2018**, *35*, 136–147. [CrossRef]
10. Lee, N.Y.; Kim, Y.; Sang, Y. How do journalists leverage Twitter? Expressive and consumptive use of Twitter. *Social Sci. J.* **2017**, *54*, 139–147. [CrossRef]
11. Kiritchenko, S.; Zhou, X.; Mohammad, S.M. Sentiment Analysis of Short Informal Text. *J. Artif. Intell. Res.* **2014**, *50*, 723–762. [CrossRef]
12. Bravo-Marquez, F.; Pfahringer, B.; Frank, E.; Mohammad, S.M. Affective Tweets: A Weka Package for Analyzing Affec in Tweets. *J. Mach. Learn. Res.* **2019**, *20*, 1–6.
13. Hutto, C.J.; Gilbert, E. VADER: A Parsimonious Rule-Based Model for Sentiment Analysis of Social Media Text. *AAAI Assoc. Adv. Artif. Intell.* **2014**. Available online: www.aaai.org (accessed on 26 June 2019).
14. Saura, J.R.; Bennet, D. A Three-Stage Methodological Process of Data Text Mining: A UGC Business Intelligence Analysis. *Symmetry* **2019**, *11*, 519. [CrossRef]
15. Twitter. Standard API Rate Limits Per Window. 2018. Available online: https://developer.twitter.com/en/docs/basics/rate-limits (accessed on 23 June 2018).
16. Internet Live Stats. Twitter Usage Statistics. 2018. Available online: https://www.internetlivestats.com/twitter-statistics/ (accessed on 23 June 2018).
17. TextBlob. TextBlob. 2018. Available online: https://textblob.readthedocs.io/en/dev/ (accessed on 23 June 2018).
18. NLTK. Natural Language Toolkit. 2018. Available online: http://www.nltk.org/ (accessed on 21 June 2018).
19. University of Antweerpen CLiPS. Pattern. En. 2018. Available online: https://www.clips.uantwerpen.be/pages/pattern-en (accessed on 23 June 2018).
20. Twitter. API Reference Index. 2018. Available online: https://developer.twitter.com/en/docs/api-reference-index.html (accessed on 23 June 2018).
21. Tweepy. Tweepy. 2018. Available online: http://www.tweepy.org/. (accessed on 23 June 2018).
22. MySQL-Python. MySQLdb User's Guide. 2018. Available online: http://mysql-python.sourceforge.net/ (accessed on 23 June 2018).
23. Matplotlib. Matplotlib. 2018. Available online: https://matplotlib.org/ (accessed on 23 June 2018).
24. Martin, C. Syria Attack: Nerve Agent Experts Race to Smuggle Bodies out of Douma. 2018. Available online: https://www.theguardian.com/world/2018/apr/12/syria-attack-experts-check-signs-nerve-agent (accessed on 23 June 2018).
25. Julian, B.; Peter, B. Syria: US, UK and France Launch Strikes in Response to Chemical Attack. 2018. Available online: https://www.theguardian.com/world/2018/apr/14/syria-air-strikes-us-uk-and-france-launch-attack-on-assad-regime (accessed on 23 June 2018).
26. Ferrara, E. Disinformation and social bot operations in the run up to the 2017 French presidential election. *First Monday* **2017**. [CrossRef]