Sentiment Analysis in social media towards government measures against COVID-19

Jeroen Ritmeester
Student Nr.: 1838369
University of Twente
Enschede, Netherlands
j.c.ritmeester@student.utwente.nl

Nils Rublein
Student Nr.: 1864432
University of Twente
Enschede, Netherlands
n.rublein@student.utwente.nl

ABSTRACT

In this paper the effect of governmental measures towards COVID-19 on the mass sentiment in social media has been investigated. Four different classifiers have been compared against each other, namely Naive Bayes, Logisitic Regression, VADER and FastText. No significant difference could be observed between the classifiers, as they all yielded a similar performance of ca. 77% in terms of accuracy and F1 scoring. Based on geo-tagged tweets from the U.S., the sentiment over time towards COVID-19 has been visualized and compared with real-life events. Two different data sets have been successfully used to automatically classify the average sentiment over time and visualise it, showing features in the time plot that may correspond to real-life events surrounding the virus.

KEYWORDS

Natural Language Processing, Sentiment Analysis, COVID-19, Social Media

1 INTRODUCTION

A wide variety of opinions are voiced online with regards to the fight against the Coronavirus by the leaders of nations worldwide. In the Netherlands, for example, there is a large ongoing public debate about the validity and effectiveness of measures instated by the government, like the upcoming obligation to wear mouth masks everywhere in public areas, cancellation and postponing of all events, and shutting down of restaurants, bars and hotels. There are also people who are content with these measures and agree that they are necessary evils. This split in sentiment regarding the measures does not only occur in the Netherlands, though. Thanks to social media, one can easily find this divide in sentiment in almost any language that the user can read. To get a better understanding of this split, we aim to classify virus related tweets for their sentiment, and track the change in sentiment over time. Making use of geolocation embedded in the tweets, we can then create a timeline and clarify visible changes in sentiment by finding the measure or event that occurred at the date that shows said change. Not only does this give an interesting insight in how the public responds to certain events, but understanding people's thoughts on their government and its policies can also help to find out how such policies should be introduced and communicated. In this case, the focus will be on the United States of America, because of its intense interaction with the virus, and because English tweets are far more abundant than Dutch ones.

In essence, we want to analyse how sentiment changes over time in respect to the easing / hardening / introduction of measures against COVID-19 by the national government of the U.S.A. (using social media). Thus, the main research question is formulated as follows:

"How do government regulations related to COVID-19 affect the sentiment over time on social media?"

To answer this question more precisely, the following sub-questions will be investigated:

- What preprocessing steps are required to be able to classify sentiment based on tweets?
- What classifier is best suited to predict sentiment based on tweets?
- To what extent correspond the classification to actual sentiment and real-life events?

The remainder of this paper is structured as follows: First, in section 3, a method to answer the research question is presented. This is followed up in section 4, where the corresponding results are shown. Then, in section 5, the results are discussed and interpreted. Finally, in section 6 a conclusion is given by summarising the main findings of this paper, answering the research question and discussing the possibility of future improvements.

2 RELATED WORK

This is most certainly not the first time Twitter is used for sentiment analysis, more particularly regarding the virus. However, the topic is in this case not very relevant with regards to the performance of the classifier; any abundant topic can be mined from Twitter using their API and a structured approach to distill the topics from the tweets themselves. As such, there is a good amount of information available in earlier performed research. For example, Jiangiang and Xiaolin [1] compared four different models based on various types of text preprocessing. Useful aspects are the correction of clitics in negations (won't –> will not, don't –> do not, et cetera) and expanding of acronyms and internet language. However, the removal of URLs, numbers and stopwords as a minimal effect on the performance of the classifiers. They also did not hurt the performance either, however, so they may still be included as a form to reduce possible sources of unexpected noise.

In the work by Samuel et al. [2], the workings of various classifiers within COVID-19 Twitter sentiment analysis is explained. Among these classifiers are the Naive Bayes classifier (NB) and Logistic Regression (LR) classifier. NB is supposed to be able to achieve higher performances on short tweets, and the LR performs well assuming the data points are independent from each other. LR does tend to be less stable according to the authors. The paper also stated that the performance drops as the Tweets get longer. However, the research used only one sentiment lexicon, even though they state that "it's a useful exercise to evaluate comparatively with other sentiment lexicons".

Finally, the paper by Bao et al [3] uses a dataset 1.6 million tweets that have been labelled automatically, using the emotions as noisy labels. Because of the size and positive/negative balance (800,000 positive tweets, 800,000 negative tweets) in this data set, this is a good input for us to train our own classifiers on. This paper reiterates some things that were mentioned earlier in the work of Jiangiang and Xiaolin [1], namely that negation transformation and repeated letter normalisation improves the performance of their model. One difference is that the authors state here that URL removal also improves performance. Another important text preprocessing aspect is stemming and lemmatisation, which showed a negative impact. This is interesting to verify for ourselves, as lemmatisation is supposed to reduce the feature space and thus allow for higher accuracies.

3 METHOD

This section describes the data being used, how the data is preprocessed and which classifiers are being used to analyse the sentiment.

3.1 Data

For this project two data sets have been used. To analyse the sentiment in social media over time with respect to COVID-19 regulations introduced by the government of the U.S., the geo-tagged tweets data set by R. Lamsal [4] has been used. This data set contains 246,117 tweet IDs and respective sentiment values that have been automatically created using TextBlob. All tweets are written in English. The data set has been converted from tweet IDs to the actual tweet objects that contain information such as the text of the author, the date or the location. This has been done by using `twarc` as hydrator (using the ID to retrieve the data) in combination with the Twitter API. Afterwards, the hydrated tweets have been filtered for the location "USA", which resulted in 104,023 tweets. However, when converting the sentiment values that have been obtained via TextBlob from a continuous range of $[-1, 1]$ to categorical values (positive, neutral and negative) using equally sized bins, it becomes apparent that this data set is highly unbalanced as 78% are neutral tweets, 20% positive labels and only 2% negative, as can be seen in figure 1. To balance this data set, the negative and positive tweets would need to be down sampled to the same size as the negative tweets leaving the data set at 6% (6,241 tweets) of its original size.

In order to train and compare the classifiers for sentiment analysis, a second data set with sufficient size has therefore been used, namely the `sentiment140` data set created by A. Go et al. [5]. As mentioned earlier, this data set contains 1.6 million tweets that have been automatically labelled as either positive or negative using emoticons. Each class occurs 800,000 times, which renders the data set evenly balanced.

3.2 Preprocessing

The data has been preprocessed as follows:

- Transforming tweets to lowercase,
- Removal of Twitter handles (e.g. retweets),
- Removal of user names,
- Removal of URL links,

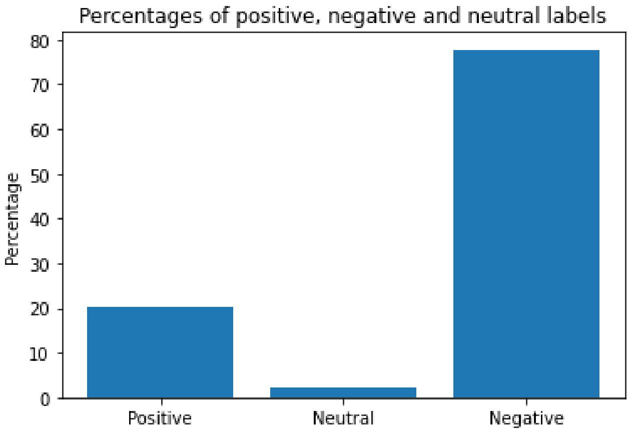

Figure 1: Distribution of classified sentiment in tweets, filtered for geolocation USA.

- Removal of punctuation,
- Correction of spelling mistakes,
- Automated tagging of part-of-speech,
- Lemmatisation of each word based on POS.

In addition, we will evaluate the classifiers with and without stopwords as well as with and without lemmatization.

3.3 Classification

To analyse the sentiment of the geo-tagged tweets, we first compare four categories of classifiers against each other. These are Naive Bayes as probabilistic classifier, Logistic Regression as linear discriminative classifier, Valence Aware Dictionary for Sentiment Reasoning (VADER) as rule-based classifier, and finally FastText as embedded classifier. These classifiers are trained and evaluated first on the `sentiment140` data set. The best classifier will then be used to analyse the sentiment on the geo-tagged data set. In the sub sections below, each classifier will be briefly described.

3.3.1 VADER. Vader is a pre-trained model that uses a lexical approach also called rule-based sentiment analysis and does thus not need any training data. Vader has been chosen as rule-based classifier as it is specifically made for social media. For instance, VADER does take into account whether upper case letters are used to emphasize a statement. As an example, "This trip was HORRIBLE!" would have a higher impact than "This trip was horrible!". In a similar manner, punctuation marks are treated the same way where more punctuation marks have a higher impact on the sentiment than less punctuation marks. In addition, VADER has also functions incorporated that handle emojis, slang phrases, as well as acronyms. As output VADER returns the probability of a given input being positive, negative, and neutral. Together these values add up to 100%. Furthermore, a compound value is being returned which normalises the sentiment to a single one dimensional measure. However, one significant drawback for rule-based sentiment classifiers is that only the individual words are being used to compute the sentient and that context around these words is ignored.

3.3.2 FastText. FastText is an extension of `word2vec` that allows to learn word embeddings, which can be trained supervised and unsupervised using the context around words. Instead of using individual words as input to the neural network, FastText splits up each word as character n-grams in addition to the word itself. For instance, with $n = 2$, the word `cat` would be split into `<c, ca, at, t>`. After learning a Skipgram embedding for each n-gram, the word `cat` is then being represented as the sum of all embeddings. This allows to retain the semantic meaning of a word that may occur in form of n-grams in other words and to understand suffixes and prefixes. An advantage of this is that the meaning of rare words can be more accurately predicted as n-grams of these words occur likely also in other words.

3.3.3 Naive Bayes. Naive Bayes (NB) are well known in the domain of automated classification. They are statistically based models that aim to calculate the probability of a data point - in this case a vectorised string - belonging to a certain class. The probability can be calculated for every possible class, and the class with the highest probability for that given data point gets assigned to it. However, this assumes that all tweets are independent from one another. This assumption may or may not be valid, but for this research, we will temporarily accept the assumption.

There are plenty of out-of-the-box NB classifiers than can be implemented. Because they are rather simple to implement, the choice was made to test each of them with the same fixed parameters and select the best one as the representative for the NB classifiers. This means the same length of n-grams, vectorisation method, token processing, and stopword filtering.

After selecting one NB model, this one can be trained on the same data set as the other classifiers will be trained.

3.3.4 Logistic Regression. Initially it was planned to use a Support Vector Machine (SVM) as literature suggests that - at least up until recently - SVMs could achieve near-state-of-the-art performance on sentiment analysis and other NLP tasks, this made for a logical decision. However, the SVMs were very time consuming to train due to their very large computational demand in relation to the other classifiers. Due to time constraints, we opted to investigate the possibility of implementing a Logistic Regression (LR) model instead. The idea is that the LR model might be good in this classification task, since the vectorised strings form may form clusters in n-dimensional space. If the model is able to draw a (n-1)-dimensional decision boundary, it could very well be a potent replacement for the SVM. Since there are a lot of hyperparameters that can be set which may drastically alter the operation of the model, we will use an automated grid search to find the best hyperparameters based on a relatively small sample (n = 40,000). These hyperparameters will then be fixed for further investigation of the model's performance. During initial investigation and familiarisation with the models, it looked like the LR model was also quite time-intensive in its training, albeit significantly less than the SVM. Therefore, we may have to train the final model on a subset of the data.

3.3.5 Parameter variations. Table 1 shows an overview of the parameters that we want to investigate for each model. The removal of stopwords alters the grammatical structure of sentences, if not destroys it. Therefore, we expect to see a difference in performance for classifiers that rely on part-of-speech tagging (POS). The token processing may be done using lemmatisation, stemming, or nothing. Lemmatisation reduces the words to a basic form, based on the word's POS, whereas stemming simply removes characters to attempt to achieve the same thing. Since token processing may result in less sparse vectorised strings, this may affect the performance of the models. In some of the works discussed in section 2, lemmatisation was found to be have a negative impact on the performance. Nevertheless, since we will be altering some other hyperparameters, we want to verify for ourselves if this will be the case. The vectorisers `CountVectoriser` and `TfidfVectoriser` work vastly different, the former simply counting the occurrence of a token, and the other giving a metric of a token's uniqueness, and therefore an indication of its importance. Since tweets are short fragments, a relatively large portion may consists of common words. To ensure this, both vectorisers will be tested. Finally, the length of the n-grams that the classifiers use can be altered. Here, unigrams, bigrams, and trigrams will be tested.

First, the focus will be on fixing the n-gram length. Once this is fixed to one of three numbers, the total number of variations is reduced to 32. After this, the vectorisers will be tested and fixed, further reducing the number of variations to 20. Then, the choice of lemmatisation versus stemming will be made, and only one of these methods will be implemented based and compared to no token processing, after which the number of variations will be 8. Finally, the choice to filter stopwords will be fixed, leaving one optimal version of each type of classifier. The classifier with the best performance will then be used for the rest of the research. For the sake of brevity, not all results may be elaborated upon numerically. Therefore, please note that this does not mean that the fixation of one of these variables was done arbitrarily.

4 RESULTS

4.1 Hyperparameters

Note: some graphs show $n = 800,000$ and others show $n = 1,600,000$. These both mean $n = 1,600,000$, since the others accidentally displayed the number of samples **per class**.

Before evaluating each classifier, some classifiers needed some prior investigation with regards to hyperparameters. First we needed to determine what version of the NB classifier we wanted to use. To achieve that, a multinomial NB, Bernoulli NB, and complement NB were all trained on $n = 40,000$ data points, all with n-gram lengths of 1, 2, and 3. The results of all these runs are shown in Appendix 6. As is shown, the highest accuracy and F1 score is achieved by the Bernoulli classifier using unigrams. This means that this classifier will represent the NB contender. For every NB classifier tested here, it can be observed that the longer the n-grams are, the lower the performance is. This also held for trials with bigger data sets. Based on this result, we fixed the n-gram length for all classifiers, i.e. NB and the other classifiers, to 1. This choice reduces the amount of variations to 32.

For the LR classifier, we used the `GridSearchCV` function to automatically generate the best hyperparameters, based on a run with $n = 40,000$ data points. This yielded the following: `penalty="l2"`, `C=0.2338`, `solver="liblinear"`, and `max_iter=1000`. These settings are used for all subsequent evaluations of the LR classifier.

Finally, for FastText, there is an `autotune` function that searches similar to the `GridSearchCV` function

Variable property	VADER	FastText	Naive Bayes	Logistic Regression	Variation factor
Filter stopwords	X	X	X	X	2
Lemmatisation/Stemming/None		X	X	X	3
TFIDF/Count vectoriser			X	X	2
n-gram length		X	X	X	3
Total variations per classifier	2	18	36*	36	Total = 92

Table 1: All possible variables that were taken into account. Variation factor indicates how the amount of possible combination multiplies. *This number is for the final NB type.

for the best hyperparameters, however we were unable to run this function successfully. Therefore we tested various hyperparameters, obtaining with `lr=0.1`, `epoch=20`, `dim=20` the best results. These parameters have been used for all subsequent evaluations of the FastText classifier.

The vectorisers `CountVectoriser` and `TfidfVectoriser` work in very different ways. However, when running a test on the entire data set, there was no significant difference between the use of either one, shown in figure 7. Therefore, we can only base our assumption for now on the intuitive idea that `TfidfVectoriser` works in a more sophisticated manner, and fix that choice. Regardless of the reasoning, the performance will not be largely affected by this choice. This brings down the amount of variations down to 20. Plots of these results can be found in Appendix 6

Next, we aimed to fix the choice between lemmatisation and stemming. We opted for lemmatisation with automated POS tagging, since lemmatisation aims to maintain at least some grammatical structure and readability, whereas stemming works in a cruder fashion. Because lemmatisation requires automated POS tagging, the expectation was that this would reduce the performance somewhat, creating more noise and errors in the process. On the other hand, the hope was that this sacrifice would be made up for the reduction of feature space, thus improving the performance in the classification phase. However, similar to the vectoriser choice, the results were barely distinct based on a data set size of $n = 80,000$, shown in figure 9. Because the spelling correction required for lemmatisation and the lemmatisation itself took over 8 hours to process 80,000 tweets, we determined against the use of token processing. This choice further reduces the amount of variations to 8.

Finally, the choice of filtering the stopwords yields barely any difference as well. This difference was tested on the entire dataset, $n = 1,600,000$, shown in figure 8. Again, since the difference is negligible and the filtering costs extra processing time, we opted against the use of stopword filtering.

With this final choice, each classifier type has now one model that can be compared against one another. In figure 2 and 3, the final result is shown. Each classifier (where applicable) used an n-gram length of 1, made use of the `TfidfVectoriser`, used no stopword filtering, and no lemmatisation or stemming. It can be observed that none of the classifiers performs significantly better than another one with exception of the VADER model, which can be confidently discounted on accounts of its accuracy and F1-score. Since the classes are perfectly balanced by our definition, it does not matter if the highest value of accuracy or F1 is used as an indication of optimal performance. Therefore, the highest score out of both of them is $F1 = 77.96$ for the LR model. Therefore, this is the model that will be used for the analysis of changing sentiment and comparing it to real-world events to find a qualitative correlation.

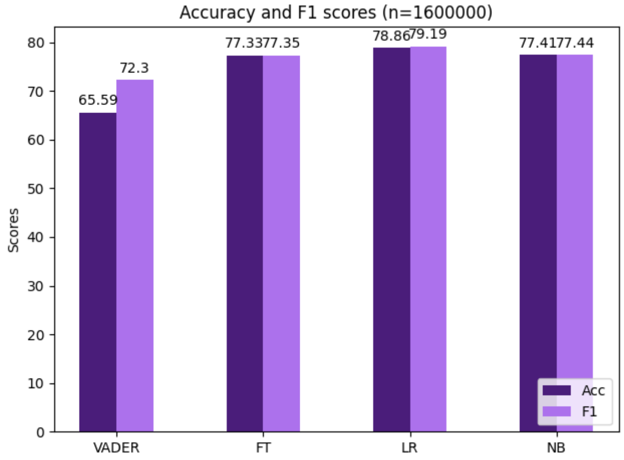

Figure 2: Bar plot for accuracy and F1 scores for each classifier with final parameters.

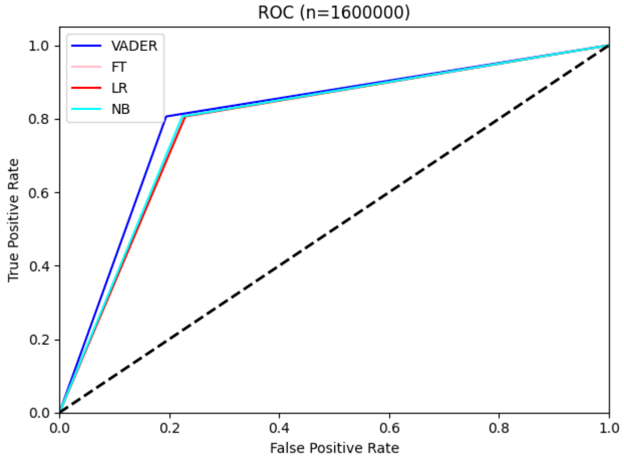

Figure 3: ROC curve for each classifier with final parameters.

Using the LR classifier as final model to predict the sentiment values for geo-tagged corona tweets and summing the sentiment values per day, the normalized sentiment over time can be seen in figure 4:

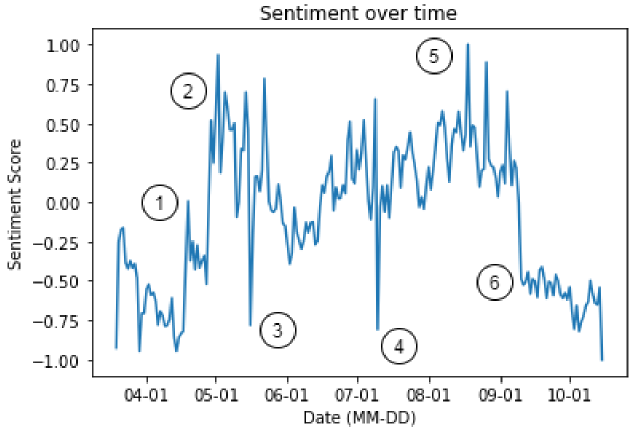

Figure 4: Sentiment over time for corona related tweets with the geo-tag USA.

where the circled numbers represent potentially relevant government measures and events that might have impacted the sentiment. These are as follows:

(1) **April 16:** Trump discusses “Gating Criteria” as a Way to reopen the US economy in time for Easter Sunday. Broad guidelines are published for how people could return to work and public venues [6].

(2) **April 29** NIH Trial shows early promise for Remdesivir [6].
May 1: "Shortly after the trial data are published, FDA grants an EUA to Remdesivir after preliminary data from an NIH trial found the treatment accelerated recovery in individuals with advanced COVID-19 and lung involvement." [6].

(3) **May 12:** Fauci testifies that the US death toll of 80,000 is likely an underestimate. He warns against the relaxation of social distancing and states that a vaccine may be effective and achieved within 1 or 2 years. [6].

(4) **July 6:** The U.S. submits its formal notification to withdraw from the WHO. [7].

(5) **August 12:** "The U.S. government secures 100 million doses of Moderna’s candidate vaccine, mRNA-1273, which is in late-stage clinical trials. The deal also gives the U.S. option to purchase an additional 400 million vaccine doses" [7].

(6) **August 31:** Confirmed cases of COVID-19 surpass 6 million in the United States, the highest number of any country [7].
September 8: AstraZeneca pauses COVID-19 vaccine trails for savety reviews after one volunteer became suddenly ill [8].
September 9: The U.S. announces it will stop screening international arrivals for COVID-19 [8].

To what extent these government measures and events are related to the actual mass sentiment is discussed in the next section.

5 DISCUSSION

In this section the choice of the data set(s), the preprocessing, (the classifiers) and their effects on the final result are being discussed.

5.1 Data set

The data set that was used to train the classifiers has been created automatically. While manual labelled data might be more precise and can handle complex sentences, it is dependent on a high inter-rater reliability rate in order to be consistent and is very expensive in time and in resources for large amounts of data. As a great deal of data was required in order to train and compare multiple classifiers, the choice fell on automatic labelling. However, automatically labelled data does also has it flaws next to its merits, as the quality of the labelling is often lower than manually annotated labels. In this case, the automatic sentiment labels have been purely based on the counts of negative / positive emoticons which is obviously a very simplified method to generate sentiment values. As all classifiers in this project have been trained on this data, their real-life performance might be lower than figures 2 and 3 suggest. In addition, there might be a wide range of unknown words in the test set (the geo-tagged data set) as the training data set is already ten years old and language changes rapidly over time in social media.

However, even though that the training data is already more than ten years old and that it has been been generated with a very simplified method, it can still be used to give an indication of the general sentiment trend. When comparing the sentiment obtained by the LR classifier with a pre-trained model such as TextBlob, it can be observed that predicted sentiment values follow the same trend, as shown in figure 5:

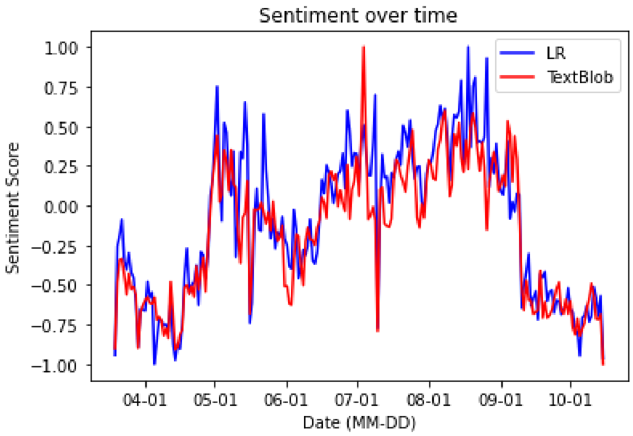

Figure 5: Sentiment over time calculated with logistic regression and TextBlob.

5.2 Preprocessing

Several preprocessing steps have not been explored in this report, including

- expanding clitics, acronyms, abbreviations
- transforming slang words
- segmenting hashtags (e.g. `#coronavirusdisaster` to `corona, virus, disaster`)

Employing these steps would decrease the sparsity of the training corpus and potentially result in higher performance.

5.3 Classifiers

The low performance of the VADER classifier might be caused by the fact that all punctuation marks have been removed from the data as well as the transformation of all letters to lowercase; these are all attributes that VADER actually uses to infer about the sentiment of a given input sentence. Further (literature) research is required for us to clearly explain the extreme similarity in the performance of the other models. We had not expected that any two models would be so close, much less so for three models.

5.4 Final sentiment analysis

As shown in figure 5 training the logistic classifier on sent140 data set seemed to be reasonable. However, sentiment values obtained via Twitter alone are not a reflection of general mass sentiment in a nation or even in a state or local area [2]. In addition, the provided government measures and events in section 4 do of course not share a one-to-one causal relationship with the predicted sentiment values, as these values are dependent on a plethora of factors. Nonetheless, the obtained data can be used to give an indication of the general sentiment trend.

6 CONCLUSION

This study was set out to investigate sentiment in social media towards government measures against COVID-19. The rapid spread of the corona virus is holding the world in its grip, creating a strong need for discovering efficient and effective strategies to interpret the flow of information and and the development of mass sentiment in pandemic scenarios [2]. The wide spread use of social media, readily available data, and the exponential growth in advancements corresponding to machine learning and natural language processing make it possible to analyse public sentiment. Understanding people's thoughts on their government and its policies can also help to find out how such policies should be introduced and communicated. Therefore, this study sought out to answer the main research question:

> *"How do government regulations related to COVID-19 affect the sentiment over time on social media?"*

In order to answer the main research question, the following sub-questions have been investigated:

What preprocessing steps are required to be able to classify sentiment based on tweets?
Several preprocessing steps have been implemented in this project, such as removal of stopwords, lemmatisation, removing of usernames or transforming letters to lowercase. During the evaluation of these preprocessing steps it became apparent that the removal of

stopwords and lemmatization do not significantly improve the performance of the classifiers. In addition, the `CountVectoriser` and `TfidfVectoriser` have been compared to each other, but no significant difference could be observed. Furthermore, when analysing different ngrams ($n = 1, 2, 3$), it was observed that unigrams performed the best as in comparison to bi- and trigrams.

What classifier is best suited to predict sentiment based on tweets?

In this report four different classifiers have been compared, namely Naive Bayes, Logisitic Regression, VADER and FastText. Each classifier (where applicable) used an n-gram length of 1, made use of the `TfidfVectoriser`, used no stopword filtering, and no lemmatisation or stemming. It has been observed that none of the classifiers performs significantly better than another one with exception of the VADER model, which exerted lower performance in terms of accuracy and F1 score. To properly answer this question, a wider range of classifiers needs to be investigated.

To what extent correspond the classification to actual sentiment and real-life events?

When plotting the calculated sentiment over time for geo-tagged tweets coming from the U.s., it could be observed that peaks coincide with measures of the government of the U.S. against corona, and other relevant events surrounding the U.S.. However, it must be emphasized that these do not share a causal relationship with the predicted sentiment values, nor does sentiment values obtained from twitter display the true sentiment of a whole nation. Nonetheless, the obtained data can be used to give an indication of the general sentiment trend and the attitude towards a certain topic.

Based on the answers to these questions, we cannot give a clear answer to the question how government regulations related to COVID-19 affect the sentiment over time on social media. However, we can be confidently state that we were successful in visualising at least some important events surrounding the virus, using our own sentiment analysis classifiers. For further research, more preprocessing methods may be explored as well as more sophisticated classifiers as considered in this report. In addition, to capture a more accurate portrayal of the true sentiment of a wider population, the extracted sentiment of multiple social media may be combined.

REFERENCES

[1] Z. Jianqiang and G. Xiaolin, “Comparison research on text pre-processing methods on twitter sentiment analysis,” *IEEE Access*, vol. 5, pp. 2870–2879, 2017.

[2] J. Samuel, G. N. Ali, M. Rahman, E. Esawi, and Y. Samuel, “Covid-19 public sentiment insights and machine learning for tweets classification,” *Information*, vol. 11, no. 6, 2020. [Online]. Available: https://www.mdpi.com/2078-2489/11/6/314

[3] Y. Bao, C. Quan, L. Wang, and F. Ren, “The role of pre-processing in twitter sentiment analysis,” *Lecture Notes in Computer Science*.

[4] R. Lamsal. Coronavirus (covid-19) geo-tagged tweets dataset. [Online]. Available: https://ieee-dataport.org/open-access/coronavirus-covid-19-geo-tagged-tweets-dataset

[5] A. Go, R. Bhayani, and L. Huang. Senitment140 data set. [Online]. Available: https://www.kaggle.com/kazanova/sentiment140

[6] AJMC. A timeline of covid-19 developments in 2020. [Online]. Available: https://www.ajmc.com/view/a-timeline-of-covid19-developments-in-2020

[7] J. L. Ravelo and S. Jerving. Covid-19 — a timeline of the coronavirus outbreak. [Online]. Available: https://www.devex.com/news/covid-19-a-timeline-of-the-coronavirus-outbreak-96396

[8] C. Kantis, S. Kiernan, and J. S. Bardi. Updated: Timeline of the coronavirus. [Online]. Available: https://www.thinkglobalhealth.org/article/updated-timeline-coronavirus

APPENDIX A

```
{
  "BernoulliNB": {
    "1": {
      "Accuracy": 0.7656406249999999,
      "F1": 0.767639650134933,
      "Precision": 0.7718029232552054,
      "Recall": 0.7635210512366947
    },
    "2": {
      "Accuracy": 0.7128875000000001,
      "F1": 0.734477775851107,
      "Precision": 0.7916962605759287,
      "Recall": 0.6849725627176385
    },
    "3": {
      "Accuracy": 0.57450625,
      "F1": 0.6844995620559734,
      "Precision": 0.9202272812231318,
      "Recall": 0.5449132282627944
    }
  },
  "ComplementNB": {
    "1": {
      "Accuracy": 0.764209375,
      "F1": 0.7620312169829471,
      "Precision": 0.7526759124269498,
      "Recall": 0.771622010027784
    },
    "2": {
      "Accuracy": 0.70778125,
      "F1": 0.713479427878075,
      "Precision": 0.7253747523457067,
      "Recall": 0.7019679484860544
    },
    "3": {
      "Accuracy": 0.565878125,
      "F1": 0.6410649276156174,
      "Precision": 0.7729056857687563,
      "Recall": 0.5476481624544752
    }
  },
  "MultinomialNB": {
    "1": {
      "Accuracy": 0.764175,
      "F1": 0.7619236909020929,
      "Precision": 0.7523394764058664,
      "Recall": 0.771755247079275
    },
    "2": {
      "Accuracy": 0.707796875,
      "F1": 0.7134569947995991,
      "Precision": 0.7252563767086588,
      "Recall": 0.7020354008985918
    },
    "3": {
      "Accuracy": 0.5658875,
      "F1": 0.6410698967516563,
      "Precision": 0.7729056857687563,
      "Recall": 0.5476554153680437
    }
  }
}
```

APPENDIX B

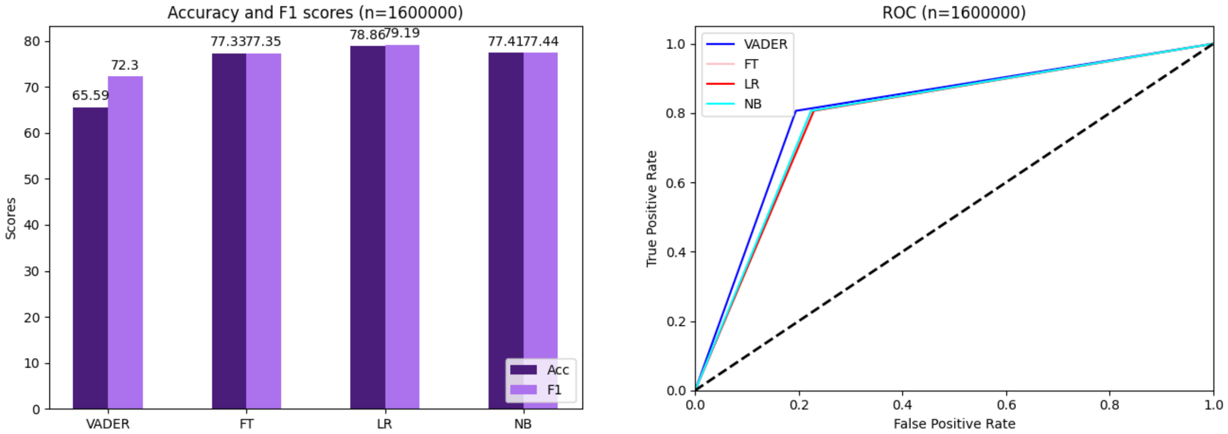

Figure 6: The performances with n-gram length of 1, using the `TfIdfVectoriser`, no stopword filtering, and no token processing.

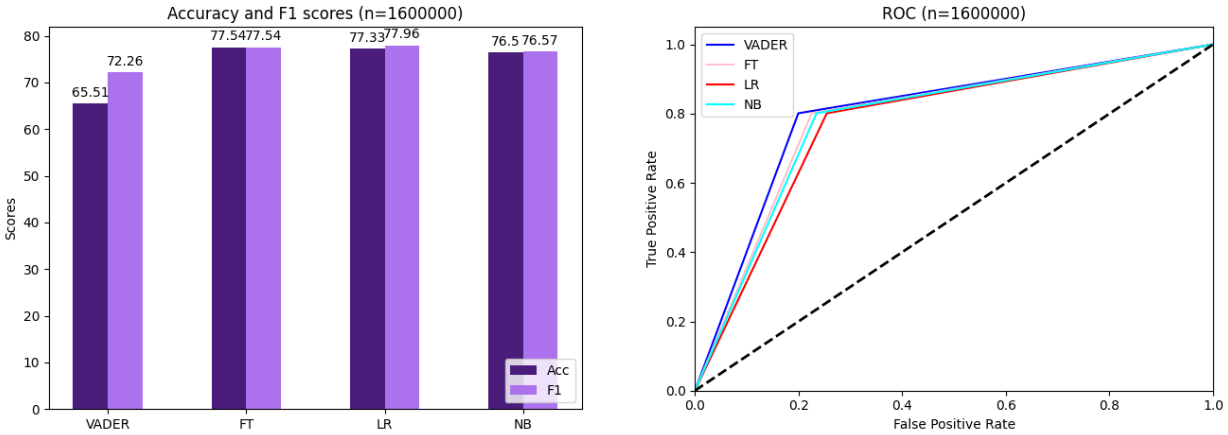

Figure 7: The performances with the same settings as figure 6, but with the `CountVectoriser`.

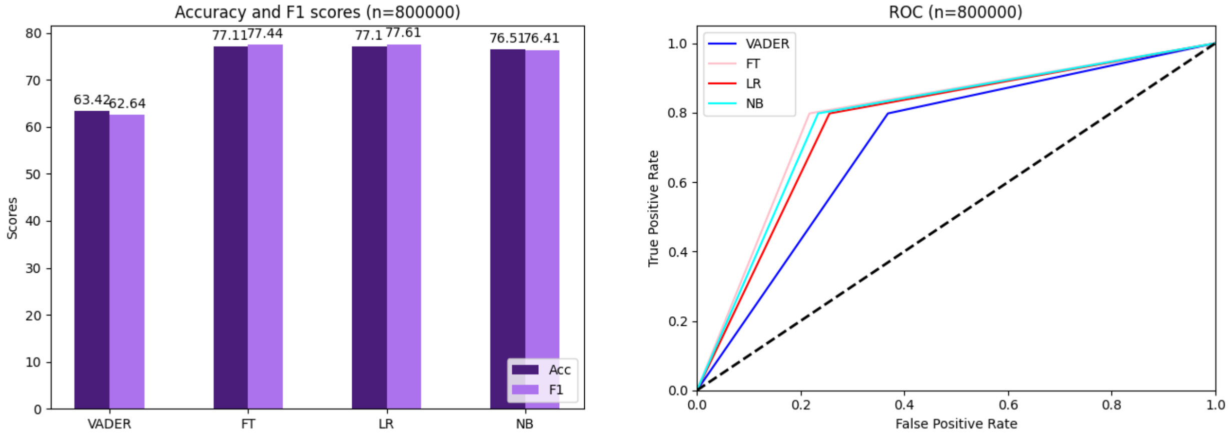

Figure 8: The performances with the same settings as figure 6, but with stopword filtering enabled. Note that $n = 800,000$ should say $n = 1,600,000$

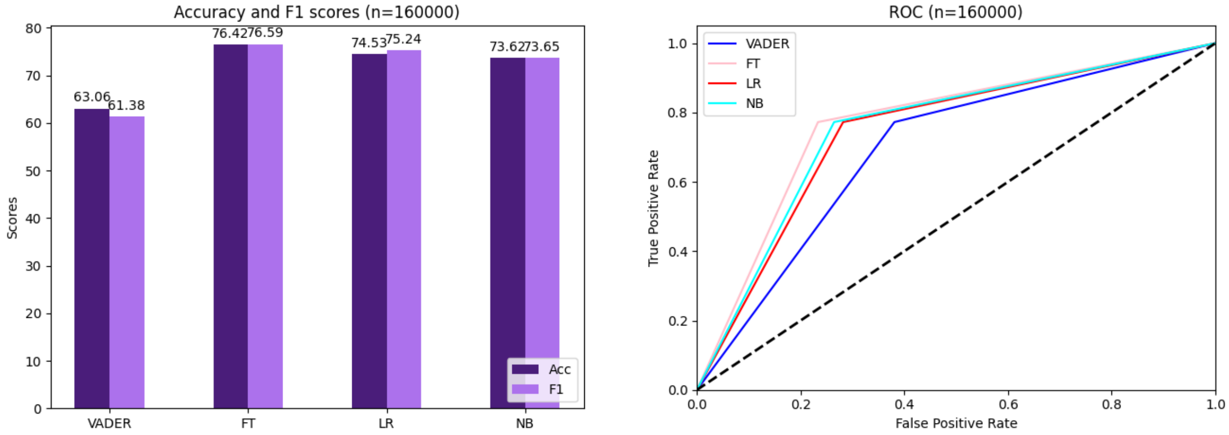

Figure 9: The performances with the same settings as figure 6, but with spelling correction and automated lemmatisation enabled. Note that this used only 10% of the data set due to time limitations.

Figure 10: Slightly frustrated statement regarding the fact that all classifiers more or less yield the same performance.