Toxic Comments Detection and Classification

Emmanuel Abraham and Thomas Chukwuka Ebere

Department of Electrical and Computer Engineering, University of Waterloo, 200 University Ave W, Waterloo, ON N2L 3G1

Abstract. The current trend of commenting behind devices (phones, laptops, tablets) has led to the propagation of multiple inciting and harmful comments that would have been harder to vocalize in face-to-face conversations. Also, this trend has created ample room for bullying, racist commenting, LGBTQ hate, sharing of abusive media content and the likes, leading to high numbers of depression and suicide. This has led to the need for social media platforms to monitor and at extremes, censor hateful speech on their platforms. However, this can be a gruesome manual task and gives place to a highly subjective interpretation of hateful and toxic speech. This project aims to use Natural Language Processing (NLP) to detect hate speech on social media platforms. Multiple algorithms would be trained to detect toxic tweets and then such can be deleted. A comparison of some algorithms would also be communicated in this paper.

Keywords: Hate speech · Natural Language Processing · Classification.

1 Introduction

Hate speech, as defined in a recent survey[1], refers to any communication that creates denigration of individuals or groups based on their race, gender, religion, ethnicity/nationality, sexual orientation or any other distinguishing characteristic (Nockleby, 2000).

It is mostly common in communications where there are perceived differences, i.e., any characteristic that makes an individual stand out differently from his or her peers. These characteristics may include race, socio-economic status, gender, sexuality, physical appearance, and behaviors.[3]

Micro-blogs, like Twitter, Facebook and a host of others, have become a common online space for discussions, trends, reviews and various forms of interaction. However, as with physical interaction, online interaction also creates avenues for individuals to make all sorts of comments and share varied opinions, based on their understanding, environment and beliefs. There are a lot of positive interactions, however, occasionally, some comments, messages and statements are clearly toxic. There are also that at first glance, seem as harmless but upon greater review, are exposed to be hateful. This review is usually done by an analysis of the sentiments shared or when the recipient reports such message as harmful.

Using a list of three statement examples excerpted from Schmidt and Wiegand (2017):

- (a) Go fucking kill yourself and die already useless ugly pile of shit scumbag.
- (b) Hope one of those bitches falls over and breaks her leg.
- (c) Put on a wig and lipstick and be who you really are.

Reviewing the statements above, (a) and (b) below are glaringly toxic and can be easily spotted because of the use of violent words. Without any context, (c) would pass as not toxic. However, if the recipient of the message is confirmed to be an adolescent male in these times going through a transition on gender, this would be a toxic comment.

The above example shows that the nature of a message can be highly dependent on the present world interpretations. Therefore, it is intuitive that the detection of hate speech finds more clarity by including information on varied aspects and not exclusively based on language.[1]

2 Literature Review

Social distancing, stay-at-home orders and the enforcement of lock-down has brought a great deal of uncertainty to the world stage. Tensions have increased across many (if not all) nations and have led to a number of cases of domestic violence, racial issues and conspiracy theories being circulated. This has been no help to the online space, an environment already experiencing rapid growth, with a rise in hate speech, racist comments and sexist ideologies. Thus, it has become of great importance to moderate online content and ensure that while many are glued to their devices and readily commenting and sharing, the information provided does not lead to further agitation.

However, the manual moderation of content on online platforms is a Herculean task, even with the most professional annotators. An estimate on one of such online platforms, the popular micro-blog Twitter, shows that there were 6,000 tweets per second[5] (as at 2014) and about 9,090 tweets per second[6] as at May 2016. This shows the high volume of online traffic that would need to be reviewed for hateful speech. This would require a large number of annotators and be very expensive to manage. For example, the Amazon Mechanical Turk platform, a crowd-sourcing service used to coordinate the use of human intelligence for tasks that computers are currently unable to do, can be applied to only about several hundreds to few thousands of tweets[9]. Manual annotation may also have some bias to the annotator's subjective predictions.

This led to an interest in a search for a solution using machine learning techniques. A number of research papers have highlighted a number of methods attempted using Natural Language Processing ranging various levels of granularity from the document level to the sentence level and then to the the phrase level.

In Turney[7] (2001), an unsupervised learning algorithm called PMI-IR, which uses Point-wise Mutual Information (PMI) to analyze statistical data collected by Information Retrieval (IR). It was implemented for the recognition of synonyms, such that given a problem word and a set of alternative words, the algorithm would select the word closest in meaning to the problem word. However, the algorithm's performance was dependent on the size of the document collection that is indexed by the search engine and its query language[7]. This algorithm was later applied to a semantic review classification implementation in Turney[9] (2002), where it was used to estimate the semantic orientation of phrases and classify them as either recommended or not. In Blei et al.[2] (2003), Latent Dirichlet Allocation (LDA), a probabilistic document model, was introduced with the goal of finding short descriptions of the members of one collection that would enable efficient processing of larger collections while preserving the essential statistical relationships that can be used for classification and other tasks[2].

Go et al.[15] (2009) , who implemented a solution using sentiment analysis as a binary classification, classifying tweets as either positive or negative. As it was difficult to manually tag the sentiment of tweets at the time, they used distant supervision in building their machine-learning classifier and implemented the Multinomial Naïve Bayes (MNB), Maximum Entropy (MaxEnt) and SVM (Support Vector Machine) classifiers. They also made use of emoticons as noisy labels to differentiate between negative and positive tweets. Their results showed the NB as the most effective method with bigrams as features, achieving accuracy of 82.7% accuracy. However, the method showed a bad performance when applied against a three-class problem ("negative", "positive" and "neutral"). Pak and Paroubek[16] (2010) also used emoticons as labels, however, their implementation was on a different set of machine learning algorithms: SVM, the Conditional Random Field (CRF) classifier, and the Multinomial Naïve Bayes (MNB). They also implemented the solution as a multi-class classification, attempting to classify about 300,000 tweets as either positive, negative or neutral. Their results observed the MNB as best performer with n-grams and parts-of-speech (POS) tags.[8]

Advancements in machine learning led to the application of deep learning methods to solve image recognition and natural language problems. Likewise, researchers ventured into the Deep learning space to provide solutions. Dong et al.[17] (2014) proposed an Adaptive Recursive Neural Network (AdaRNN) using a dependency tree in order to find the words syntactically related with the target and to propagate the sentiment from sentiment words to the targets. This was evaluated on a manually annotated data set consisting of 6248 training and 692 testing tweets and managed to obtain an F1 score of 65.9%[8]. Their annotated data set was then used in future work by other researchers. One of the methods used in this paper would use a form of Recurrent Neural Network (RNN), the LSTM neural network architecture. In Wang et al.[20] (2015), a bi-directional LSTM (BLSTM-RNN) neural network architecture was used in tagging. It was implemented using an open source GPU-based toolkit of BLSTM-RNN, called CURRENT, with the activation functions of input layer and hidden layers being logistic functions, while the output layer used a soft-max function for multiclassification. The neural network was trained using statistical gradient descent algorithm with constant learning rate. This was one of the pioneering papers to show an effective way to use BLSTM-RNN for dealing with various NLP tagging tasks[20].

Malmasi and Zampieri[4] (2017) worked on a multi-class algorithm using a linear SVM classifier with two feature groups (surface n-grams, and word skip-grams) to classify texts across three labels - Hate speech, Offensive speech (but not hateful) and Acceptable speech ("OK"). In Kunal et al.[10] (2018), the Naïve Bayes (NB) algorithm was used to access and classify live Twitter reviews and tweets. It was implemented using Tweepy and TextBlob python libraries and achieved accuracy of about 92.58%.

Due to the availability of suitable text collections, a majority of studies on offensive and hateful language, including ours, have been based on data in the English language. However, more recently, a few studies have investigated using detection in other languages. Mubarak et al. (2017) addresses abusive language detection on Arabic Twitter (offensive, vulgar and hate speech)[11] and Su et al.[12] (2017) presented a system to detect such in Chinese. Some other annotations for other languages include the annotations for socially unacceptable online discourse as in Fiser et al.[18] (2017) in Slovene (language in Slovenia) and for the German language as shared in the data set of about 500 tweets in Ross et al.[13] (2016) and the pilot of the GermEval Shared Task on the Identification of Offensive Language as in Wiegand et al.[14] (2018) which had an annotated data set of over 8,000 tweets. These aim to create opportunities for future work in other languages.

The variety of context in English words, however, may reduce the accuracy of the prediction of the model created if tested with sarcastic comments and ironical statements, where meanings may be misconstrued. Also, for better comparability, access to more data sets from social media platforms would aid accuracy, but there are limitations of data and user privacy.

3 Implementation

3.1 Data

When selecting the data set, live data from Twitter was one of the choices, however, the wait for Twitter API keys caused a delay and other options had to be considered. The data set used for this work was gotten from Kaggle[19]. It is a popular data set for Hate Speech issues, extracted from Twitter data, with 24,783 tweets. It has three labels highlighted below:

- Hate Speech,
- Offensive Language,
- Neither or Normal.

3.2 Data Preprocessing

Extraneous columns: We removed extraneous columns from the data set to reduce possible noise and also to focus on the columns with textual data.

Stop words and non-letter data: To further clean the data, we removed stop words and non-alphabetic characters like exclamation marks (!) that precede words and a number of others.

Usernames: Tweets may sometimes contain other usernames by user mentions (for example @mleew17, which mentions another user with the Twitter handle). This was also cleaned out.

Repeated Letters for emphasis: We converted all letters to lowercase and split the words, performing stemming on all the words to take them to their root form. Tweets are known to contain very casual language, for example, if you search for the word "happy" with an arbitrary number of a's, p's, or y's in the middle (e.g. haaapppy, haaapppppppyyy) on Twitter, there will most likely be a nonempty result set. Thus words like these were returned to their root form of "happy".

Count Vectorization: We performed count vectorization on the words to give each word a numerical representation.

Splitting the Data: We then split the data into test and train set using a random seed of 42 with 20% of the data used for testing.

3.3 Learning algorithms Used

A number of models were used in the implementation:

K-means clustering: as defined in the project proposal, this was the algorithm to be used as a novel approach to toxic comment classification by training the data

Naïve Bayes (NB): this algorithm was selected for this paper because of its simplicity and how well it works with text categorization.

K-Nearest Neighbor (KNN): This was used because of its singular tuning parameter and because it provides good accuracy values. We also implemented the algorithm to compare with the K-means algorithm

Decision Tree classifier (DTC): this algorithm was selected because of its versatility of application and the number of hyper-parameters to be tuned is almost null.

Random Forest classifier (RFC): this algorithm was selected because it generally provides a high accuracy and is able to handle large data sets.

LSTM (Long Short Term Memory): This is a form of the Recurrent Neural Network (RNN) and was selected because of the RNN advantage of considering the sequence of data and its ability, unlike feed forward neural networks, to deal with the vanishing gradient problem.

4 Results and Analysis

4.1 K-means Clustering model

This was the proposed method of choice in our proposal but its performance on the test set was abysmal with an accuracy of 13.68%. Its predictions were largely neutral with the Normal class having the highest tally in predictions, with the Offensive categories and Hate speech classes tallying around 100 in second and third respectively. (see Fig. 1 below). This is probably due to the fact that the clusters in this data set are of varying sizes and density and its high dependence on the initial values. Because clustering is unsupervised, no "truth" is available to verify results. The absence of truth complicates assessing quality.

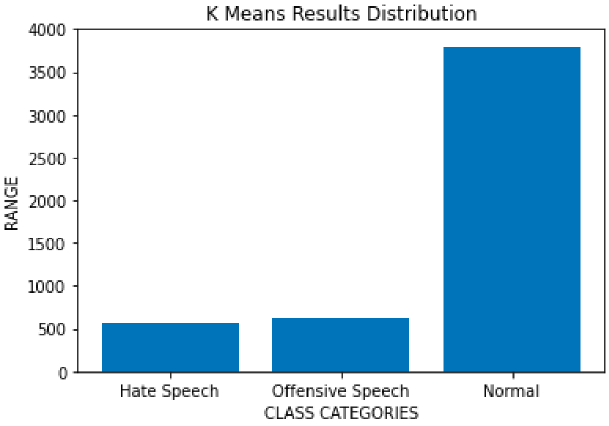

Fig. 1. Distribution of the K Means clustering model

4.2 Naïve Bayes (NB) model

On the NB model, we observed an accuracy of 48.50%. The distribution of the test set shows that Hate speech had the highest tally in predictions with Offensive language and the Neither classes slightly tied (see Fig. 2 below). We presume that the weakness of this classifier is due to the multi-class nature of the data and the assumption of independent predictors.

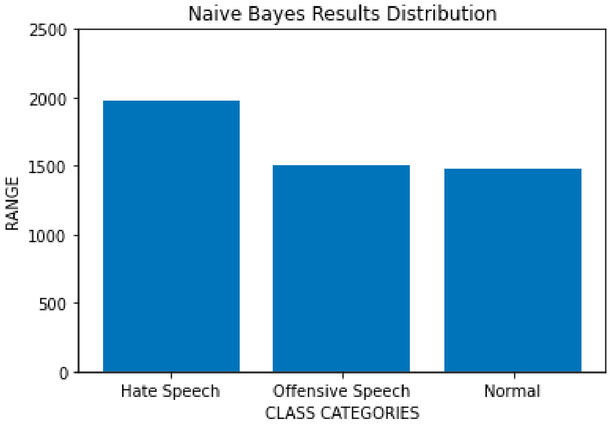

Fig. 2. Distribution of the Naïve Bayes classifier

4.3 K-Nearest Neighbours (KNN) model

Using the KNN model, we obtained a classification accuracy of 85.01%. The distribution of the test set shows that the Offensive language had the highest tally with the Normal class in second and then the Hate speech class (see Fig. 3 below). The performance of the classifier can be attributed to its strength in dealing with adapting to new data especially when a good value of K is selected.

4.4 Decision Tree Classifier (DTC)

The DTC had an accuracy of 85.29%, slightly better than the KNN model. Offensive Language had the highest tally followed by Normal class and then, Hate speech (see Fig. 4 below). Tree models are well suited to multi-class problems and this influences the relatively higher performance of the decision tree and the random forrest classification methods.

4.5 Random Forest Classifier (RFC)

The RFC performed better than the other models above with accuracy of 87.59%. The Hate Speech class distribution was well below 300, while that of Offensive Language well above 3700 and the Normal class with one slightly below 1500 (see Fig. 5 below).

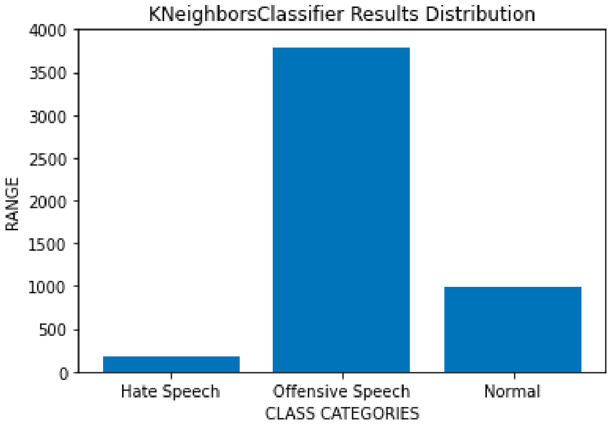

Fig. 3. Distribution of the KNN classifier

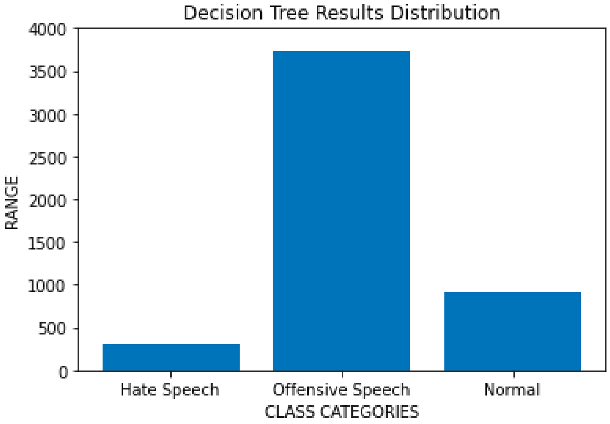

Fig. 4. Distribution of the Decision Tree classifier

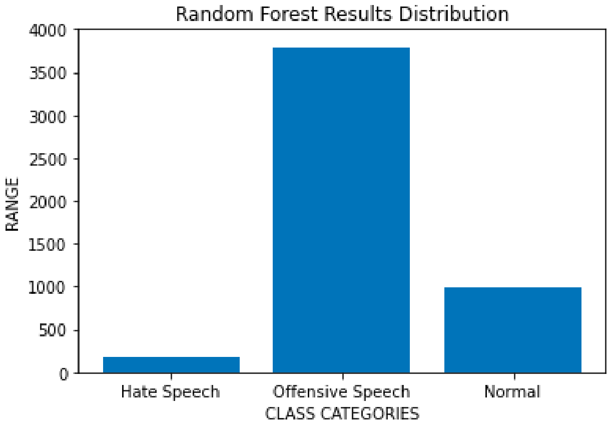

Fig. 5. Distribution of the Random Forest classifier

4.6 Long Short Term Memory (LSTM) Recurrent Neural Networks (RNN) architecture

Repeated runs of the algorithm improved its accuracy ranging from 79.90% to 80.24% to 83.75% and and its running time ranged between 22 minutes to 18 minutes to 16 minutes. However, the process was highly computationally expensive. This proves useful because of the application of word embedding, where the words are mapped to real value vectors and common words are assigned a similar vector value. A plot showing the difference between predicted output and the actual output. (see Fig. 6 below)

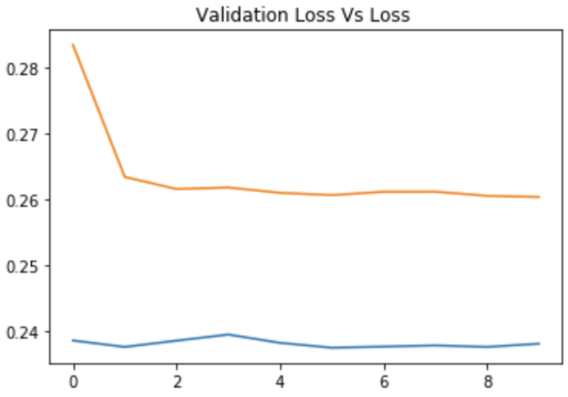

Fig. 6. Comparison of the Losses in the LSTM-RNN

Table 1 gives a summary of the accuracy and the time taken to fit of the algorithms that were implemented in this paper.

Table 1. Results Summary.

Classifier	Accuracy (%)	Time taken to fit
K-means Clustering	13.68	1 loop, best of 3: 1.95 s per loop
Naive Bayes (NB)	48.50	10 loops, best of 3: 35.4 ms per loop
K-Nearest Neighbours (KNN)	85.01	1 loop, best of 3: 681 ms per loop
Decision Tree Classifier (DTC)	85.29	1 loop, best of 3: 317 ms per loop
Random Forrest Classifier (RFC)	87.59	1 loop, best of 3: 3.37 s per loop
LSTM (Long Short Term Memory)	83.75	18 minutes

5 Conclusions and Future Work

One of our biggest lessons from this paper was the failure of the clustering method selected (K-Means clustering) to give high performance. Understanding the flaws made and improving on them would be part of the future work for the authors of this paper. We would attempt to batch the data into balanced cluster sizes and use those mini-batches. Also, the use of the LSTM recurrent neural network architecture in the field of social media moderation seemed to produce great results and would be improved on. We would aim to properly tune the network to obtain better accuracy and a shorter run-time as this is a viable area that would provide more opportunities for controlling the propagation of hate speech on social media.

References

1. A Schmidt and M Wiegand. 2017. A Survey on Hate Speech Detection Using Natural Language Processing. In Proceedings of the Fifth International Workshop on Natural Language Processing for Social Media. Association for Computational Linguistics. Valencia, Spain, pages 1–10.
2. D M Blei, A Y Ng, and M I Jordan. 2003. Latent Dirichlet Allocation. Journal of Machine Learning research 3(Jan): 993–1022.
3. J Xu, K Jun, X Zhu, and A Bellmore. 2012. Learning from bullying traces in social media. In Proceedings of the 2012 conference of the North American chapter of the association for computational linguistics: Human language technologies. Association for Computational Linguistics, pages 656–666.
4. S Malmasi and M Zampieri. 2017. Detecting Hate Speech in Social Media. arXiv:1712.06427v2 [cs.CL] 26 Dec 2017
5. David Sayce https://www.dsayce.com/social-media/tweets-day. Last accessed 11 August 2020
6. Internet Live Stats https://www.internetlivestats.com/one-second/tweets-band. Last accessed 12 August 2020

7. P. Turney, 2001. Mining the Web for synonyms: PMI-IR versus LSA on TOEFL. Proceedings of the Twelfth European Conference on Machine Learning (pp. 491-502). Berlin: Springer-Verlag
8. A Giachanou and F Crestani. 2016. Like it or not: A survey of Twitter sentiment analysis methods. ACM Comput. Surv. 49, 2, Article 28 (June 2016), 41 pages.
9. P. Turney. 2002. Thumbs up or thumbs down? semantic orientation applied to unsupervised classification of reviews. ACL.
10. S Kunal, A Saha, A Varma, V Tiwari. 2018. Textual Dissection Of Live Twitter Reviews Using Naive Bayes. International Conference on Computational Intelligence and Data Science (ICCIDS 2018) https://doi.org/10.1016/1234567890
11. H Mubarak, A Rashed, K Darwish, Y Samih, A Abdelali. 2017. Arabic Offensive Language on Twitter: Analysis and Experiments. arXiv:2004.02192v2 [cs.CL] 18 May 2020.
12. H Su, Z Huang, H Chang and C Lin. 2017. Rephrasing Profanity in Chinese Text. Proceedings of the First Workshop on Abusive Language Online, pages 18–24, Vancouver, Canada, July 30 - August 4, 2017. c 2017 Association for Computational Linguistics.
13. B Ross, M Rist, G Carbonell, B Cabrera, N Kurowsky, M Wojatzki. 2016. Measuring the Reliability of Hate Speech Annotations: The Case of the European Refugee Crisis. 10.17185/duepublico/42132.
14. M Wiegand, J Ruppenhofer, M Siegel. 2018. Overview of the GermEval 2018 Shared Task on the Identification of Offensive Language. Proceedings of GermEval 2018, 14th Conference on Natural Language Processing (KONVENS 2018), Vienna, Austria – September 21, 2018. - Vienna, Austria: Austrian Academy of Sciences, 2018. Pp. 1-10
15. A Go, R Bhayani, and L Huang. 2009. Twitter Sentiment Classification Using Distant Supervision. Technical Report. Stanford.
16. A Pak and P Paroubek. 2010. Twitter as a corpus for sentiment analysis and opinion mining. In Proceedings of the 7th on International Language Resources and Evaluation Conference (LREC'10). European Language Resources Association (ELRA), 1320–1326.
17. L Dong, F Wei, C Tan, D Tang, M Zhou, and K Xu. 2014. Adaptive recursive neural network for target-dependent twitter sentiment classification. In Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers). Association for Computational Linguistics, Stroudsburg, PA, 49–54.
18. D Fiser, N Ljubešić, T Erjavec. 2017. Legal Framework, Dataset and Annotation Schema for Socially Unacceptable Online Discourse Practices in Slovene. Proceedings of the First Workshop on Abusive Language Online, pages 46–51, Vancouver, Canada, July 30 - August 4, 2017. Association for Computational Linguistics.
19. Hate Speech and Offensive Language Dataset — Kaggle, https://www.kaggle.com/mrmorj/hate-speech-and-offensive-language-dataset. Last accessed 12 July 2020
20. P Wang, Y Qian, F K Soong, L He, H Zhao. 2015. A Unified Tagging Solution: Bidirectional LSTM Recurrent Neural Network with Word Embedding. arXiv:1511.00215v1