Replication of Detecting Abusive Language on Twitter

Aboli Moroney, Harini Ramprasad, Mayank Goel, Samarth Modi

## Introduction

Lately, there has been a lot of effort and research on identifying content that is abusive or offensive on online and social media. Twitter recently published a relatively large and reliable dataset on ‘Hate and Abusive Speech on Twitter’. As data science students, we fully understand the need to find the best methods and data for identifying such content and flagging it as inappropriate.

In this project, our aim is to replicate some of the findings in a research paper titled ‘[Comparative Studies of Detecting Abusive Language on Twitter](https://arxiv.org/abs/1808.10245)’ (see Lee, Yoon, and Jung 2018) that performs a comparative study and provides suggestions for using different algorithms and additional data features for improving such classification of hate and abusive speech using Twitter data.

By leveraging the data and code provided by the authors [Git Repository](https://github.com/younggns/comparative-abusive-lang/blob/master/README.md), we aim to replicate the efficacy and accuracy for the Logistic Regression model. Our ultimate goal is to understand how this model classifies different types of tweets and generate similar results for precision, recall and f1-score as reported by the authors. The choice of model to replicate is based on the observation that the top precision has been generated by Logistic Regression model in the original research as well as our experimental constraints for executing the code.

**Disclaimer:** Since this is real data collected from Twitter, some of the text and visual outputs in this document may contain offensive language. None of these are views or statements made by authors of this project and should be treated as impersonal data used for the sake of classification purposes only.

## Data Analysis and Model Replication

The [original data file](https://github.com/ENCASEH2020/hatespeech-twitter) contained 20 attributes for a collection of 14,509 unique tweets. On careful consideration, we selecetd the following 3 attributes which were relevant to our analysis.

1. **tweet\_id**: unique identifier for each tweet
2. **tweet\_text**: raw tweet (uncleaned)
3. **does\_this\_tweet\_contain\_hate\_speech**: classification of tweets with 3 different classes

Below is a snapshot of original data.

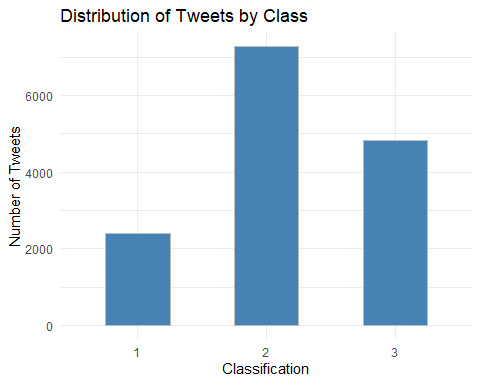
Table 1: Raw tweets from original dataset

|  |  |  |
| --- | --- | --- |
| tweet\_id | tweet\_text | does\_this\_tweet\_contain\_hate\_speech |
| 1666196150 | Warning: penny boards will make you a faggot | The tweet uses offensive language but not hate speech |
| 429512078 | Fuck dykes | The tweet contains hate speech |
| 395623778 | (**???**) (**???**) (**???**) (**???**) (**???**) at least i dont look like jefree starr faggot | The tweet contains hate speech |
| 497514685 | “(**???**):”(**???**): (**???**) Is a fag" jackie jealous" Neeeee | The tweet contains hate speech |
| 588923553 | (**???**) You heard me bitch but any way I’m back th texas so wtf u talking about bitch ass nigga | The tweet uses offensive language but not hate speech |
| 203816023 | (**???**) your a dirty terrorist and your religion is a fucking joke, you go around screaming Allah akbar doing terrorist shit. Dirty faggot. | The tweet contains hate speech |
| 1945524931 | RT (**???**): (**???**) looking like faggots? | The tweet contains hate speech |

### Tweet Classification

In this dataset, the tweets are classified into 3 classes. Below is a summary and visual representation of the distribution of tweets by classes:

## Number of Tweets  
## The tweet contains hate speech 2399  
## The tweet is not offensive 7274  
## The tweet uses offensive language but not hate speech 4836



To get a better idea of the sentiments behind tweets, we used the R package ‘WordCloud’, which generates a word cloud of the most frequent words in the given tweets.



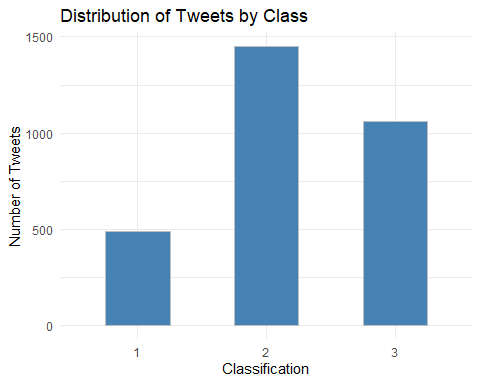
Figure 1: Wordcloud of most frequent words in tweets

### Data Pre-processing

The original dataset contains 14,509 tweets and can be found in the ‘Data’ folder as [twitter-hate-speech-classifier-data.csv](https://github.com/abolim/Reproducibility-Research-Replication/tree/master/Data) in our repository. When we executed our data processing scripts using this data, we generated a huge intermediate modeling dataset which could not be processed using our local machines due to memory and hardware constraints. Therefore for our replication study, we have taken a random sample of 3000 tweets from the original dataset, which reasonably represents the original distribution of classes.

Below is a visual representation of the distribution of tweets by classes for this sampled dataset which can be found in the ‘analysis’ folder in our repository as [sampled\_tweet\_dataset.csv](https://github.com/abolim/Reproducibility-Research-Replication/tree/master/aanalysis). The R code used to generate this dataset can be found in [001-tweet-cleaning.R](https://github.com/abolim/Reproducibility-Research-Replication/tree/master/analysis)

## Number of Tweets  
## The tweet contains hate speech 491  
## The tweet is not offensive 1450  
## The tweet uses offensive language but not hate speech 1059



As part of pre-processing, the following steps were taken to clean the ‘tweet\_text’ column:

1. Converted all strings to lower case: Using base R tolower() function
2. Removed everything that is not a number or alphabet: Using stringr package
3. Removed punctuations, special characters, twitter handles, emojis: Using stringr package
4. Removed most frequent 200 stopgap words: Using tm package

Here is the quick comparison between raw and cleaned tweets after pre-processing steps.

Table 2: Refined dataset with cleaned tweets

|  |  |
| --- | --- |
| tweet\_text | tweet\_clean1 |
| Wetback mountain dew. | wetback mountain dew |
| Last night proved that Black people can have no multi racial coalitions. It’s fuck wetbacks, spics, camel jockeys, chinks, | last night proved black people can multi racial coalitions s fuck wetbacks spics camel jockeys chinks |
| Like I’m legit scared to have sex. All these fucking diseases are real. It’s really not worth my life. | like m legit scared sex fucking diseases real s really worth life |
| (**???**) TOGETHER I LOVE THAT WORD ;) BUT SADLY , I CANT LET YOU BE A TRASH WITH ME , CAUSE YOU ARE MY EVERYTHING | together love word sadly cant let trash cause everything |
| (**???**) (**???**) (**???**) Not by much at all. I want QB to be amazing, but it’s right to be wary. | lyfe much want qb amazing s right wary |

This [refined\_tweet\_dataset.csv](https://github.com/abolim/Reproducibility-Research-Replication/tree/master/analysis) can be generated using the R code [001-tweet-cleaning.R](https://github.com/abolim/Reproducibility-Research-Replication/tree/master/analysis) and is stored in the ‘analysis’ folder.

### Tokenization and One-hot Encoding

The next step in reproducing was to create features for the model by tokenizing the tweets. In the original Python code, the authors had used the ‘nltk’ library. We tried various packages in R which provide tokenization functions and ultimately concluded that the ’quanteda’was the most suitable library to generate similar tokens as original study.

After tokenization, the next step was to encode the tokens using one-hot encoding to generate features. After one-hot encoding the output csv file was stored in the data folder as [prepared\_tweet\_dataset.csv’](https://github.com/abolim/Reproducibility-Research-Replication/tree/master/Data). This dataset contains 3,000 tweets and 7,237 features generated. This contains the one-hot encoded data for the selected 3000 tweets which is an input to run the machine learning model.

Below is a snapshot for a few rows and columns from this dataset.

Table 3: Prepared dataset with tokenized features

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Classification | dew | mountain | wetback | black | camel | can | chinks | coalitions | fuck |
| 3 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| 3 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 |
| 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

### Modeling

The above dataset can be used to bypass the resource intensive pre-processing steps. It can be used to directly train the model and reproduce the figures that we have claimed to replicate using this project. The modeling script can be found in [002-modeling.R](https://github.com/abolim/Reproducibility-Research-Replication/tree/master/analysis)

There are certain other factors to keep in mind when it comes to detecting abusive text on social media. A lot of the abusive text online tends to go undetected because of the context relative, sarcastic, subjective nature of this text. Some tweets can be classified as objectively abusive. However there are a lot of situations wherein that nuance is gone undetected. These are some glaring pitfalls of detecting abusive and hateful language online. Most neural network models are being trained to circumvent these drawbacks. However, since our model performs logistic regression, it is important to note that these factors have not been accounted for.

Our model used the Liblinear package for logistic regression. The documentation for this package stated that their given logistic model performed both L1 as well as L2 regularization for better model generalization. Our original paper utilized only L2 regularization. Additionally, the original paper used the BFGS optimization technique. However, upon further investigation into R packages, we were unable to find a library that allowed an equivalent implementation of BFGS. Since our research suggested that Liblinear was a state-of-the-art model for logistic regression, we settled for using this library.

Furthermore, our model used a direct one-hot encoding of the tokenized tweets as features to input into our logistic regression model. However, in the original paper, the authors have used both a one-hot encoding, as well as a tf-idf (term frequency-inverse document frequency) scheme of feature engineering. This was done using n-grams and fed for each classifier. Initially, it was in our agenda to also implement this methodology. However due to time constraints, we decided against this additional implementation.

## Reproducibility

### Software

The original code for the ‘Comparative Studies of Detecting Abusive Language on Twitter’ was written in language Python and also used TensorFlow. For our replication project, we have attempted to replicate the machine learning model for Logistic Regression in the R language. This involved using different packages and functions in R to process the data as well as build the model and obtain the results.

### Dataset

Additionally, due to limited computational capacity, we randomly selected 3000 tweets from a dataset of 14,509 tweets from the ‘[Hate Speech Dataset] (<https://github.com/ENCASEH2020/hatespeech-twitter>)’ provided by ENCASE. In the original research paper, the authors were able to execute the models with a dataset containing approximately 100,000 tweets. This was not necessary for us as we were not building the complex neural network models which require large amounts of data.

### Algorithm Selection

The original paper had a comparative study of 5 different machine learning and deep learning algorithms. However, for our replication purpose we have chosen ‘Logistic Regression’ model using word-level features. This choice was made as the author stated that this model outperformed all the machine learning techniques and had an F1-score which was equivalent to the best CNN model. For our project, we also had limited computational resources due to which execution of other machine learning and deep learning models was out of scope.

Further, the original paper modeled all algorithms at word and character levels separately. But, they stated that character levels models reduced the accuracy for hate and abusive speech data. Therefore, we have chosen only the word-level representation for our replication study.

## Results

Here is a quick comparison of the original figures produced by the authors against our replicated results. We observe that there is a moderate gap between the original and replicated figures.

Table 4: Original metrics

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Classification | Precision | Recall | F1.score | Accuracy |
| Hateful | 0.620 | 0.161 | 0.254 | NA |
| Not offensive | 0.807 | 0.933 | 0.865 | NA |
| Offensive, not hateful | 0.868 | 0.844 | 0.856 | NA |
| Overall | 0.786 | 0.802 | 0.780 | NA |

Table 5: Replicated metrics

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Classification | Precision | Recall | F1.score | Accuracy |
| Hateful | 0.341 | 0.423 | 0.378 | 0.75 |
| Not offensive | 0.955 | 0.851 | 0.900 | 0.75 |
| Offensive, not hateful | 0.641 | 0.701 | 0.670 | 0.75 |
| Overall | 0.646 | 0.658 | 0.649 | 0.75 |

## Conclusion

We observed lower precision, recall, and f1-scores as compared to the original study. There could be a number of different factors causing this difference in the results. Some of them are as follows:

1.We realize that since we have only taken 3,000 tweets to perform our analysis (due to limited computational resources), there is a chance for underfitting in our model. This could be a reason that our results show lower values for most metrics when compared to the original paper. 2. While converting the code repository from Python to R, some of the library functions used in R were not exactly the same as those in Python. These small differences in intermediate stages could have inflated the final difference in figures. 3. Perhaps, the original code itself is not very reproducible and has a tendency to generate different results with slight changes. 4. The original paper was done on a large scale and focused intensively on feature engineering whereas we completed feature engineering as thoroughly as we could with the available R libraries, but not to the same degree as that of the original paper.

However, we believe that we were able to successfully process the data and obtain a reasonable output using the Logistic Regression model in the limited time and resources which were available.

## Acknowledgements

We would like to thank Prof.Ben Marwick for his indiscriminate support throughout our endeavor to reproduce the results of the original paper. We are also grateful to our teaching instructor Liying Wang, who was helpful in guiding us to success in this project. We are also thankful to the developers and maintainers of the R libraries that helped us reproduce the results. Lastly, we thank our peer evaluators, who will review our work for very valuable feedback.

## References

Founta et al. (2018)

Founta, Antigoni-Maria, Constantinos Djouvas, Despoina Chatzakou, Ilias Leontiadis, Jeremy Blackburn, Gianluca Stringhini, Athena Vakali, Michael Sirivianos, and Nicolas Kourtellis. 2018. “Large Scale Crowdsourcing and Characterization of Twitter Abusive Behavior.” In *11th International Conference on Web and Social Media, Icwsm 2018*. AAAI Press.

Lee, Younghun, Seunghyun Yoon, and Kyomin Jung. 2018. “Comparative Studies of Detecting Abusive Language on Twitter.” *CoRR*. <http://arxiv.org/abs/1808.10245>.