Tweet Classification For Automated Cyber-Hate Flagging

Daniel Hammocks[1] and Emma Luk[2]

[1]City, University of London, [2] City, University of London and Professional Member of British Computer Society (MBCS)

THE PROBLEM & IMPORTANCE OF TASK

Twitter has been accused of failing to act swiftly when abusive messages have been reported to the social media giant[1], combining this with the fact that "Social media leads children to violence"[2] as stated by Met Police Commissioner, Cressida Dick, there has been a real concern from numerous groups about the lack of intervention from social media giants whose platforms are propagating crime.

The recent spike of violent crime in London, to which many young people have been involved, has highlighted the importance of tackling the root of the problem by developing tools to reduce the escalation of hate-crime in the cyberworld to physical crime in the real world.

DATASET

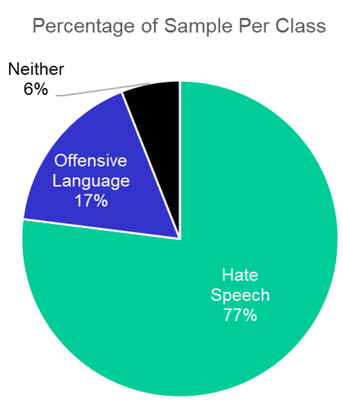

Figure 1: Proportion of data samples per class.

We used an open dataset[3] which was a pre-labelled sample of approximately 25,00 tweets which had been classified as either hate speech, offensive language or neither.

In order to identify any potential class bias we performed an initial analysis, as shown in Figure 1, to count the number of samples per class. Since only a small proportion of our data (6%) was classified as neither it would not be realistically feasible to reduce the other classes to an equivalent size.

However, due to a large proportion of our dataset being classified as hate crime we hypothesised that any potential bias would be towards this class and hence increasing our false positive rate which is ideal for the task as we would rather identify non-hate tweets than mis-predict any that do contain hate crime.

It should be noted that this is a severe limitation of this dataset and that for future work a larger set of equivalent class size should be used.

METHODOLOGY

We assess the generalisability of our models by splitting our data into a training and test set; this will be done randomly, within samples, in a 9:1 ratio respectively. We will train our models on the training set. We will also use a proportion of the training set as a validation set to enable comparison between like models when performing hyper-parameter optimisation. Finally, we will use the test set to compare the best of our two classifiers.

Every second approximately 6,000 tweets are sent, we have therefore chosen to use a Random Forest (RF) and Naïve Bayes (NB) classifier instead of a deep learning model due to their prediction speed and relatively high accuracy. For the NB classifier we will be optimising the smoothing parameter and for the RF classifier we have chosen to optimise the number of trees used.

RESULTS OF HYPER-PARAMETER OPTIMISATION

Naive Bayes

Training Size	Feature Extractor	Training Time	Smoothing	Number of Features /Vocabulary Size	Time to Predict on Train Set	Training Accuracy	Time to Predict on Test Set	Test Accuracy
14895	BagOfWords	147.354712	1	100	0.098081350	77.4891	0.08485	78.7217
22307	BagOfWords	160.4816289	1	100	0.033895969	77.4600	0.05763	78.6812
7412	BagOfWords	124.6565824	1	100	0.039091110	77.4960	0.09659	78.6408
7412	TF-IDF	68.86121011	1	500	0.067363262	77.4555	0.05602	78.2767
22307	TF-IDF	86.92750168	1	1000	0.042253733	77.6214	0.04739	78.2767
14895	TF-IDF	92.74658346	1	500	0.038824081	77.5227	0.04180	78.1553

Random Forest

Training Size	Feature Extractor	Training Time (s)	Number of Trees	Number of Features /Vocabulary Size	Time to Predict on Train Set (s)	Training Accuracy (%)	Time to Predict on Test Set (s)	Test Accuracy (%)
7412	TF-IDF	569.6227849	50	100	0.045049906	77.5420	0.10752	78.4049
14895	TF-IDF	630.8806994	50	100	0.048483133	77.2248	0.07729	78.4049
22307	BagOfWords	757.1233068	50	100	0.072860956	77.3305	0.04924	78.4049
22307	TF-IDF	649.4542322	50	100	0.076583385	77.3305	0.04046	78.4049
7412	BagOfWords	690.9166691	500	100	0.098002672	77.7166	0.04355	78.3640
14895	BagOfWords	729.2091455	250	100	0.067946434	77.2987	0.06623	78.3640

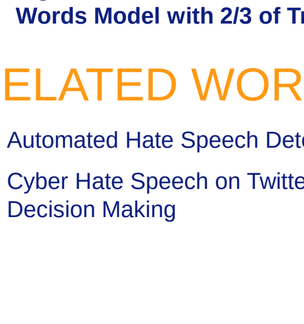

Figure 4: Box and Whisker Plot of Model Accuracies

BEST PERFORMING MODEL

Figure 5: Confusion Matrix of NB Bag of Words Model with 2/3 of Training Data.

Our best performing model was achieved using the NB classifier, the bag of words feature extractor and two thirds of the training data. NB makes for a good model as it was also able to classify our test set in less than 0.1 seconds making real time prediction a feasible task. Whilst the accuracy of our model is 78.7% looking at the confusion matrix (Figure 5) we can see that the majority of non-hate crime predictions are falsely classified as hate crime and we believe this is due to the fact that are sample sizes were unequal resulting in a biased classifier rather than due to overfitting because of a lower accuracy on the training set - highlighting a need for future work with equal sample sizes.

RELATED WORK

- Automated Hate Speech Detection and the Problem of Offensive Language.
- Cyber Hate Speech on Twitter: An Application of Machine Classification and Statistical Modelling for Policy and Decision Making

INTRODUCTION

In this project we aim to use a cluster computing framework (Apache Spark) to create a classifier which will automate detection of the presence of hate-speech, offensive language or neither of the two aforementioned classes in a short section of text. This can also be used to assess the proportion of hate-speech or offensive language an individual uses on social media.

We will analyse a selection of circa 25,000 tweets which have been manually labelled by users of the CrowdFlower platform. We will hold out a selection of data for comparing two machine learning classifiers (Multi-nomial Naive Bayes & Random Forests) as well as two feature extraction techniques (Bag of Words, and Term Frequency-Inverse Document Frequency(TF-IDF)). In order to identify the best parameters for our feature extraction, training set size, and classifiers we will use a validation set to perform hyper-parameter optimisation.

PRE-PROCESSING

We undertook a number of data processing tasks prior to applying our machine learning pipelines. Initially, we discarded any information deemed to be irrelevant for classifier, for example the number of analysts that determined a tweet as one of the three classes as our classifier would not have this information for new data.

A key part of our analysis was the twitter specific processing. Hashtags are used on twitter to identify a tweet with a particular theme; often using capital letters to distinguish between words to enable easy comprehension for the reader; for example, #CityUniversityLondon. Using this information we split hashtags on each capital letter to ensure we were not loosing information relevant to the tweets context. Twitter also uses 'handles' (@username) to distinguish who has written a tweet and if they have mentioned others. These were removed since we expect more than a small group of users to distribute hate content and for users to publish tweets both containing and not containing offensive/hate speech. As well as that we removed 'RT' tokens that identify a tweet as being content written by someone else and re-tweeted by a user as these hold no meaning to context.

As with usual natural language processing we also removed punctuation and used a twitter stop-word list[4] to remove words deemed irrelevant to a texts meaning. We also removed URLs since they are shortened and hold no meaning.

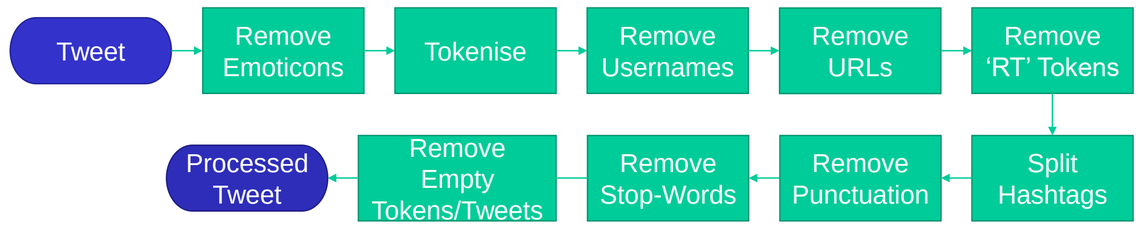

Figure 2: Figure outlining the processing stages taken for each tweet.

CLASSIFICATION PIPELINE DIAGRAM

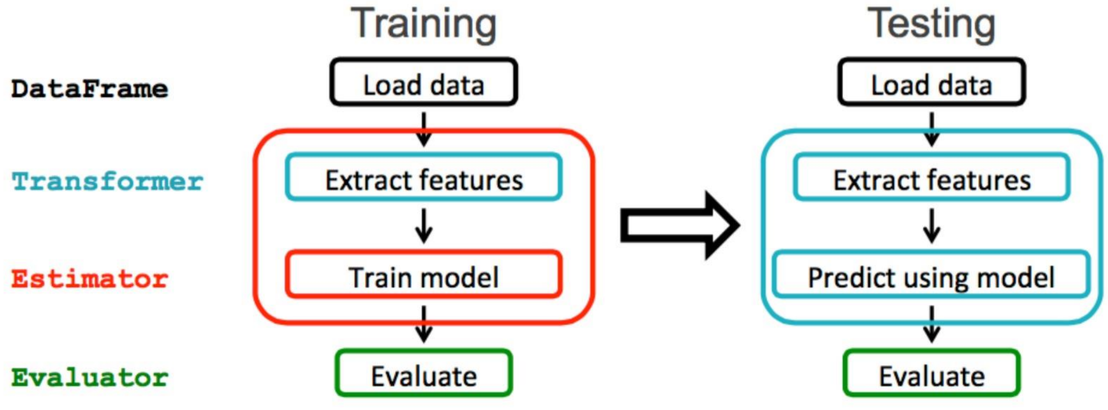

Figure 3 [5]: Showing an overview of the Spark pipeline for training and testing.

ANALYSIS & CRITICAL EVALUATION OF RESULTS

An overview of the accuracies for our two models is shown in Figure 4. We notice that the variance for the accuracies reported by a RF model is smaller than that of our NB model but that our NB model returns more accurate results. From our results we have also identified that the NB model works best when the bag of words feature extraction technique is used whereas for the RF model TF-IDF is the superior feature extraction method. Our RF models also indicate that varying the training size has no effect on the accuracy of the model whilst the NB model shows no correlation between training size and results. We also note that our RF models take on average 6 times longer to train than our NB models and that there is a direct correlation in training time and sample size.

In terms of the hyperparameters we have come to the following conclusions:

- The optimum NB smoothing value is 1; which is used in all of our NB models.
- The optimum number of trees for our RF models vary depending on the feature extractor with TF-IDF favouring less trees and the Bag of Words performing better with more trees.
- The optimum number of features/vocabulary size is consistently 100 for our random forest models. These results are echoed for our NB Bag Of Words models however, the TF-IDF favours a larger vocabulary size.

LESSONS LEARNED, CONCLUSION & FUTURE WORK

We conclude that using Naïve Bayes with a Bag of Words feature extractor was the superior method for text classification compared to the other methods evaluated. Whilst we achieve a relatively high accuracy we need to consider the effect of class bias and how this may have affected our results.

In order to further our work we have identified a few key areas for development:

- Using word dictionaries to assist in hashtag splitting where capitalisation is not used.
- Application of our models to other social media platforms (such as Facebook and Instagram)
- Creation of Deep Learning models whilst assessing the accuracy and the time taken to predict
- Increase the range of detection by attempting to identify other areas of criminal activity such as terrorism do cryptocurrency pump-and-dump scams within *terrorism and cryptocurrency

REFERENCES

[1] https://www.theguardian.com/technology/2017/aug/22/twitter-failing-to-act-on-graphic-images-and-abusive-messages-says-mp

[2] http://www.bbc.co.uk/news/uk-43603080

[3] Davidson, Thomas and Warmsley, Dana and Macy, Michael and Weber, Ingmar. Automated Hate Speech Detection and the Problem of Offensive Language. Proceedings of the 11th International AAAI Conference on Web and Social Media. ICWSM '17. 2017. Montreal, Canada. 512-515

[4] Wei, Gong. (Date Unknown). Stop words for tweets. [Available at: https://sites.google.com/site/iamgongwei/home] Bhayani, R., Go, A., & Huang, L. (2009). Twitter Sentiment Classification using Distant Supervision.

[5] http://mapr.com/blog/fast-data-processing-pipeline-predicting-flight-delays-using-apache-apis-pt-1