Hate speech detection using SVM, RFM and CNN model

Heena Khan

1 Abstract

The Indian Institute of Technology's(ISM) 2018 Workshop on Trolling, Aggression, and Cyberbully (TRAC) hosted a shared task focused on detecting aggressive text in both English and Hindi. Our paper is inspired by one of the papers introduced during this workshop 'TRAC-1 Shared Task on Aggression Identification in Social Media for COLING 2018' [10]. In our paper, we are using the same dataset used by them; our dataset is labeled as Overtly Aggressive, Covertly Aggressive, and Non-aggressive. We have trained our model for two tasks, English (Facebook) task and English (Social Media) task, which also contains multi-lingual comments. We have used a Deep Learning Convolution Neural Network, and multi-class classifiers Support Vector Machine and Random Forest Method. Our accuracy for task 1 CNN model is (0.09345) above average 10-fold validation (0.4490) and task 2 CNN model is (0.08554) above average 10-fold validation (0.4532). Our accuracy for task 1 SVM model is (0.00065) below average 10-fold validation (0.5431), task 2 SVM model is (0.01480) above average 10-fold validation (0.53447), for task 1 RFM model is (0.00277) above average 10-fold validation (0.4997), and for task 2 RFM model is (0.03513) below average 10-fold validation (0.5362)

2 Introduction

Hate speech is currently a huge problem and is growing considerably. It particularly affects marginalized communities, like minorities, women, people of color, and other vulnerable communities. There are a lot of papers introduced in the last few years that focused on the identification of abuse occurring on social media in the United States and the United Kingdom [16]. India has been equally active and working it's way up to achieve its position in social and racial abuse ranking. In 2018-19 Islamophobic content was the biggest source of hate speech on Facebook in India, accounting for 37 percent of the content reported by Equality Labs. Fake News (16 percent), casteism (13 percent), and gender/sexuality hate speech (13 percent) were the next biggest groups [6]. The comments and posts that called Bengali Muslims "pigs", "terrorists", "dogs", "rapists", and "criminals" seemingly in violation of Facebook's standards on hate speech were shared nearly 100,000 times and viewed at least 5.4 million times, showed the Avaaz review, which covered 800 Facebook posts related to the state of Assam alone. [14]. 'Facebook India is Overrun by Islamophobia' says David Gilbert from 'The Vice' [6]. The 2018 Workshop on Trolling, Aggression, and Cyberbully (TRAC) hosted a shared task focused on detecting aggressive text in both English and Hindi [10]. The papers that were introduced during this workshop focused on Facebook and Twitter comments from the same region. Though our dataset is labeled as Overtly aggressive, Covertly aggressive, and non-aggressive, It is mostly Islamophobic.

3 Background

There are 7.7 billion people in the world, with at least 3.5 billion of us online. This means social media platforms are used by one-in-three people in the world, and more than two-thirds of all internet users [15]. The number of online users has been directly proportional to online cyber-bullying, trolling, and abuse. Recent years have seen an increasing number of research on hate speech detection as well as other related areas [12]. The researchers have used different classification methodologies to identify social abuse. Davidson et al. [2] used an automated approach to identify hate speech on the USA twitter dataset using lexicon from Hatebase [7]. Their data is labeled as hate speech, offensive, and neither of them. Waseem and Hovy worked on their dataset from USA twitter consisting of 16,914 tweets labeled as racist, sexist, or neither [18]. De Gilbert et al. [3] introduced their data consisting of posts from a white supremacist forum labeled with Hate, No-Hate, Relation, or Skip category. Kaggle has introduced a shared task on detecting insulting comments, and the dataset is released to the public [8]. Hala Mulki et al. [13] introduced a dataset in Arabic for Hate speech and abusive language detection. Ross et al. [17] created a Twitter dataset in German for the European refugee crisis, labeled as hate and Not. There has been an ample number of researches been done to recognize hate speech on social media throughout America, Europe, and the Middle East. A 2018 Workshop on Trolling, Aggression, and Cyberbully (TRAC) hosted a shared task focused on detecting aggressive text in both English and Hindi in South Asia the dataset is labeled as overtly aggressive, covertly aggressive and non-aggressive. The model introduced in the TRAC paper by Kumar et al. .[10] used the LSTM model and the accuracy as around 35 percent. In this paper, we will be implementing CNN, SVM, and RFM methods using the TRAC dataset, expecting them to improve the performance.

4 Methodology

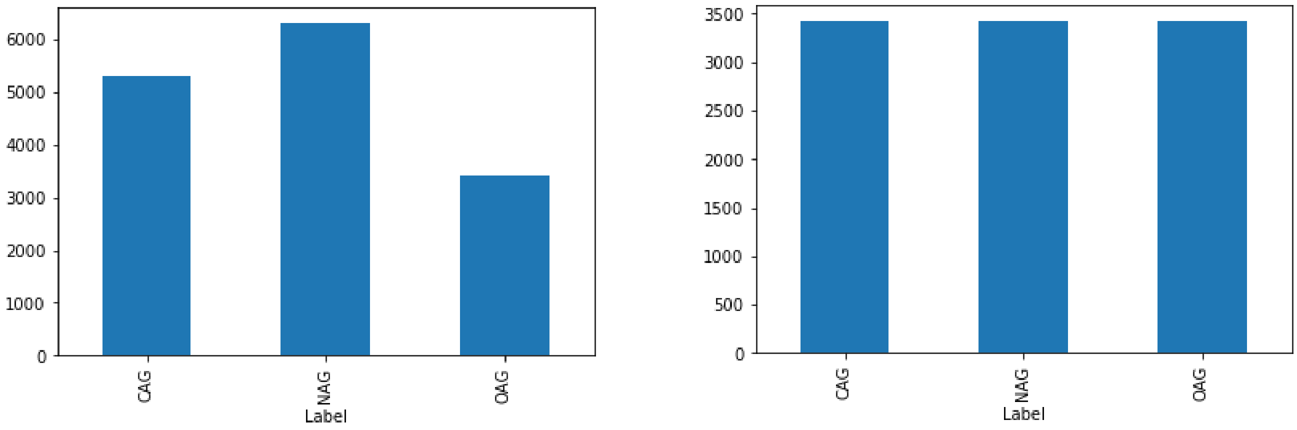
The dataset from TRAC is available to the public and contains 15,000 Facebook and around 2500 Twitter comments labeled as overtly aggressive, covertly aggressive, or non-aggressive. In our paper, we are going to create a model that can classify aggressive text using Multi-class algorithm SVM, RFM, and a Deep Learning CNN methodology. We are going to use the dataset used by Kumar, Bhanodai, Pamula, and Reddy [10]. Their dataset contains both English and Hindi text set, but we will be focusing only on the English text set. For all their runs, they have used the Long Short-Term Memory model (LSTM) and used random baseline cross-validation to measure accuracy. For Task 1 English (Facebook) dataset their best run was (i.e. 0.3572) above the random Baseline (i.e. 0.3535) and Task 2 English (Social Media) dataset their best run was (i.e.0.1960) below the random Baseline (i.e. 0.3477) [10].

The authors in TRAC [10] mentioned implementing text classification models like SVM, Random Forest method, and Deep Learning models like CNN as a classifier in their future work. We are implementing SVM, RFM, and CNN from their future work, and we will be using the English dataset only. The first step in the implementation of any model is text pre-processing. Our text pre-processing involves changing all the text to lower case, removing stop words, word lemmatization, removing emoji, removing hyperlinks, removing improper full stop, and sentence continuation (E.g ....), removing all non-alphabetic text and word tokenization. The processed text is then encoded and implemented using CNN, SVN, and RFM classification methods.

Since we have an imbalanced dataset in task 1 where around 6200 comments are non-aggressive, around 3500 are covertly aggressive, and around 4300 are overtly aggressive, SVM, RFM could produce sub-optimal results with imbalanced datasets [1]. So we took care of the imbalance issue by omitting an extra number of comments in the other two classes, before dividing our data into test and train.

4.1 Using Convolutional Neural Network (CNN) for hate speech classification

In the implementation of the Convolutional Neural Network, the most important part is converting an arbitrary length string into a fixed-length vector to use Neural Net API. There are different ways of converting words to vectors like Bag Of Words (BOW), but on using BOW, the order of the words is



(a) Task1 Imbalanced Data

(b) Task1 Balanced Data

Figure 1: Displaying solution to class imbalance problem

lost, which is too little information for our neural network. LSTM, where each character in the word is encoded into one-hot representation, is too much information for a neural network. So to find a balance between too little and too much information, we are using word embedding were each word is assigned a pre-defined set of values. This is achieved by using Word2vec. Word2vec is a pre-trained model by Google used for word embedding.

The first step is to generate feature embedding. Feature embedding for all words is constructed by using word embedding. We then decide a maximum sentence length, and we use padding to convert each comment into a fixed-length vector. This vector is then passed to CNN. Our Convolutional Neural network is inspired by Yoon Kim [9]. It consists of 4 layers, the embedding layer, the Convolutional layer, the max-pooling layer, and the soft-max output layer. The embedding matrix is passed to the embedding-layer. Four different filter sizes are applied to each comment, and GlobalMaxPooling1D layers are applied to each layer. All the outputs are then concatenated. A Dropout layer, then a Dense and then a Dropout, and then a Final Dense layer is applied. The pooling layer converts each tweet into a fixed-length vector, capturing the information from the comment/tweet. The max-pooling layer then captures the most critical factors from the comment [4]. In the output, the softmax layer calculates the class probability distributions for each comment/tweet and classify comments accordingly based on probability values.

4.2 Using Multi-class classifier SVM and RFM to classify hate speech

After the word processing and word tokenization, we used TFIDFvectorizer from the sklearn library to extract the n-gram features from each comment and weight them according to their TF-IDF values. We experiment on different values of n-grams, where n ranges from one to three; this forms our feature for the SVM and RFM model [5]. We have used linear SVM from sklearn to generate our Linear SVM model and RandomForestClassifier to generate the RFM model [11].

5 Results and analysis

We followed the proper testing protocol for the classification tasks. To estimate the skill of our model on the new data, we followed a classic 10-fold cross-validation procedure for all our models. Linear

Support Vector Machine proved the best text classification algorithms for our dataset.

The average 10-fold cross-validated results for the multi-class Support Vector Machines (SVM) model and Random Forest Model (RFM) and Convolutional Neural Network(CNN) are compared to one another. According to our analysis for task 1, SVM performed (0.0941), i.e., 9 percent better than CNN and (0.0434), i.e., 9 percent better than RFM. For task 2, SVM performed (0.08127), i.e., 8 percent better than CNN and (0.00173) poorer than RFM. Table 1 is displaying the average 10-fold score for each model in both the tasks.

10 -Fold Score	Task 1	Task 2
CNN	0.4490	0.4532
SVM	0.5431	0.5345
RFM	0.4997	0.5362

Table 1: 10-Fold Cross Validation Accuracy

We have generated false-positive rate (FPR), true positive rate (TPR) matrix to identify our precision [11]. The diagonal represents (predicted label = actual label) where we want are maximum count to be. However, there are a number of mis-classifications. Figure 2 represents the FPR and TPR matrix of all the models for task 1. Figure 3 represents the FPR and TPR matrix of all the models for task 2.

Task 1: The most correctly predicted class is covertly aggressive (CAG) and the most misclassified class in OAG for all the three models CNN, SVM, and RFM. In Figure 2, we see that the maximum number of 'NAG' comments is correctly predicted for the CNN model and SVM model but not for the RFM model.

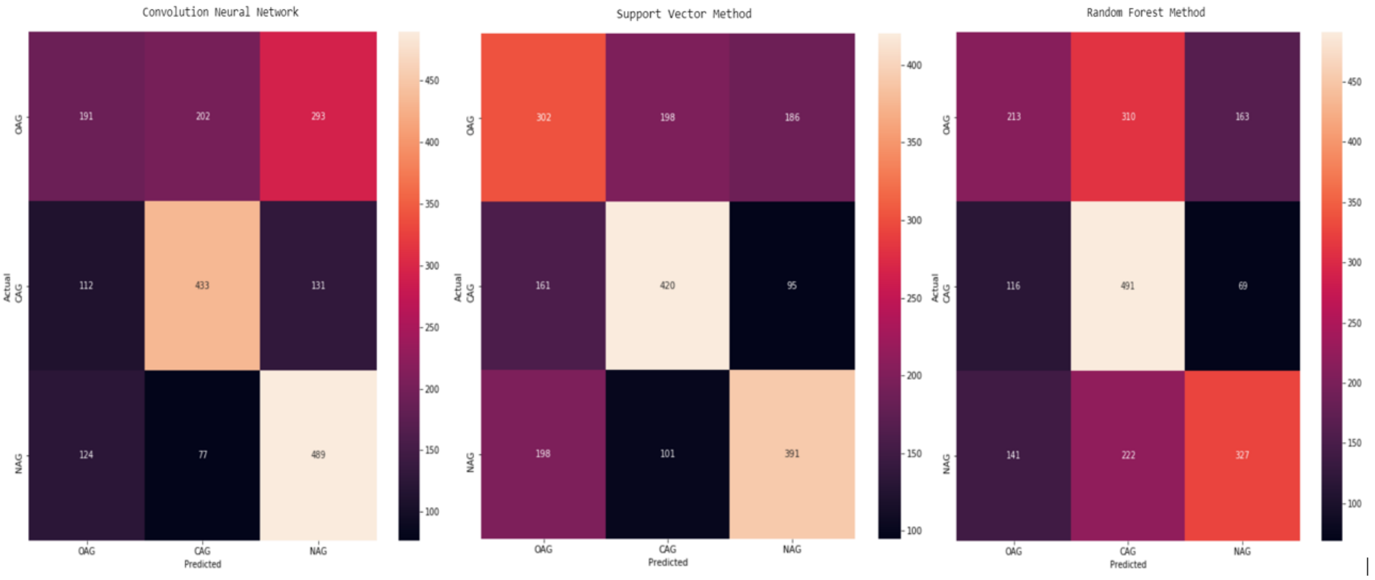


Figure 2: FPR and TPR matrix for Task 1

Task 2: Figure3, we see that the maximum number of 'CAG' comments is correctly predicted for CNN, SVM, and RFM. 'NAG' comment is predicted after 'CAG.' The lowest prediction is of 'OAG' Comment. 'OAG' class is a little bit confused with the 'NAG' class.

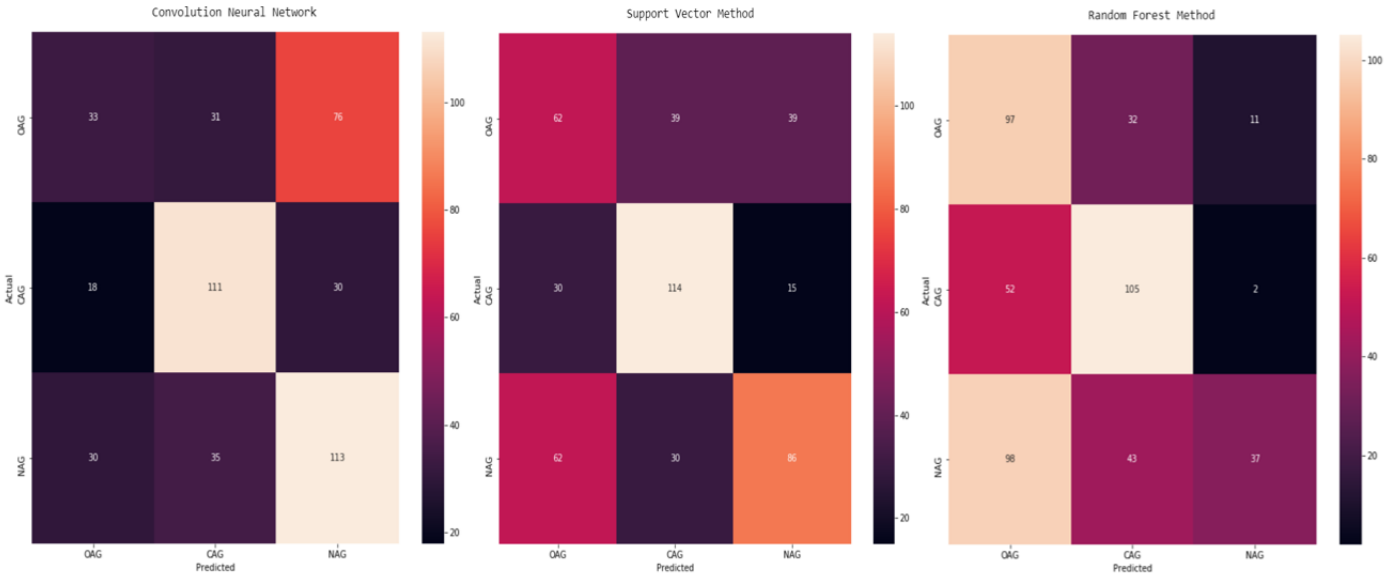


Figure 3: FPR and TPR matrix for Task 2

A confusion matrix specifying macro precision, recall, and F1 score for all three methods have been created to quantify the quality of our model. We had almost the same weighted precision, recall, and F1 score because of the balanced dataset.

Task - 1	precision	recall	f1-score	accuracy
CNN	0.53	0.54	0.53	0.54
SVM	0.54	0.54	0.54	0.54
RFM	0.51	0.50	0.49	0.50

Table 2: Confusion Matrix for Task 1.

Task - 2	precision	recall	f1-score	accuracy
CNN	0.52	0.52	0.51	0.54
SVM	0.55	0.55	0.54	0.55
RFM	0.57	0.52	0.48	0.50

Table 3: Confusion Matrix for Task 2.

6 Conclusion

According to our analysis and results, we came to the conclusion that since our dataset is consists of multi-lingual data using word embedding vectors as a feature is not sufficient. We must also use the language lexicon to generate our features. Our CNN, SVM, and RFM performed better than the LSTM model used in TRAC paper [10]. We were expecting our Deep learning CNN model would provide better results; instead, the SVM classifier yields the best result compared to CNN and RFM. We also noticed that RFM performed better on smaller datasets compared to CNN and SVM. In my future work, We would like to implement and analyze the BERT model on the same dataset. I would also like to have different feature selection methods than in the current model.

References

[1] Rukshan Batuwita and Vasile Palade. Class imbalance learning methods for support vector machines. 2013.

[2] Thomas Davidson, Dana Warmsley, Michael Macy, and Ingmar Weber. Automated hate speech detection and the problem of offensive language. In *Eleventh international aaai conference on web and social media*, 2017.

[3] Ona de Gibert, Naiara Perez, Aitor García-Pablos, and Montse Cuadros. Hate speech dataset from a white supremacy forum. *arXiv preprint arXiv:1809.04444*, 2018.

[4] Björn Gambäck and Utpal Kumar Sikdar. Using convolutional neural networks to classify hate-speech. In *Proceedings of the first workshop on abusive language online*, pages 85–90, 2017.

[5] Aditya Gaydhani, Vikrant Doma, Shrikant Kendre, and Laxmi Bhagwat. Detecting hate speech and offensive language on twitter using machine learning: An n-gram and tfidf based approach. *arXiv preprint arXiv:1809.08651*, 2018.

[6] David Gilbert. The vice - facebook in india is drowning in anti-muslim hate speech.

[7] Hatebase. Hatebase; available from: https://hatebase.org/.

[8] kaggle. Detecting insults in social commentary;. available from: https://kaggle.com/c/detecting-insults-in-social-commentary.

[9] Yoon Kim. Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882*, 2014.

[10] Ritesh Kumar, Guggilla Bhanodai, Rajendra Pamula, and Maheshwar Reddy Chennuru. Trac-1 shared task on aggression identification: Iit (ism)@ coling'18. In *Proceedings of the First Workshop on Trolling, Aggression and Cyberbullying (TRAC-2018)*, pages 58–65, 2018.

[11] Susan Li. Towards data science - multi-class text classication with scikit-learn.

[12] Sean MacAvaney, Hao-Ren Yao, Eugene Yang, Katina Russell, Nazli Goharian, and Ophir Frieder. Hate speech detection: Challenges and solutions. *PloS one*, 14(8), 2019.

[13] Hala Mulki, Hatem Haddad, Chedi Bechikh Ali, and Halima Alshabani. L-hsab: A levantine twitter dataset for hate speech and abusive language. In *Proceedings of the Third Workshop on Abusive Language Online*, pages 111–118, 2019.

[14] Casey Newton. The verge - hate speech is spreading on facebook in india again.

[15] Esteban Ortiz-Ospina. Our world in data - https://ourworldindata.org/rise-of-social-media.

[16] Marian-Andrei Rizoiu, Tianyu Wang, Gabriela Ferraro, and Hanna Suominen. Transfer learning for hate speech detection in social media. *arXiv preprint arXiv:1906.03829*, 2019.

[17] Björn Ross, Michael Rist, Guillermo Carbonell, Benjamin Cabrera, Nils Kurowsky, and Michael Wojatzki. Measuring the reliability of hate speech annotations: The case of the european refugee crisis. *arXiv preprint arXiv:1701.08118*, 2017.

[18] Zeerak Waseem and Dirk Hovy. Hateful symbols or hateful people? predictive features for hate speech detection on twitter. In *Proceedings of the NAACL student research workshop*, pages 88–93, 2016.