

Troll Detection and Anti-Trolling Solution using Artificial Intelligence/Machine Learning

Saloni Dangre^{*1}, Shubham Sharma^{*1}, Swati Balyan^{*1}, Tanisha Jaiswal^{*1}, Dr. Pankaj Agarkar², Prof. Pooja Shinde³

¹BE Scholar, Department of Computer Engineering, Dr. D. Y. Patil School of Engineering, Lohegaon, Pune, Maharashtra, India

²Head of Department, Department of Computer Engineering, Dr. D. Y. Patil School of Engineering, Lohegaon, Pune, Maharashtra, India

³Professor, Department of Computer Engineering, Dr. D. Y. Patil School of Engineering, Lohegaon, Pune, Maharashtra, India

ABSTRACT

With the increase in usage of social media platforms, due to which trolling and use of abusive language has burgeoned proportionately. The sole reason for this is that there is no surveilling authority on these platforms. Anyone from kids, teenagers to adults can fall prey to trolling. This paper focuses on using Artificial Intelligence and Machine learning algorithms to invigilate such bullies and further classify them for enhanced analysis. We will be introducing lexical, aggression, syntactic and sentiment analyzers to examine the data and determine if it was meant to be a troll or not. The output of these analyzers will be then fed to algorithms such as Naive Bayes and classifiers like Decision Tree, Random forest, Multinomial, Logistic regression to segregate the trolls in different categories like offensive, targeted, individual, group etc and use visual representation tools to improve the analysis.

Keywords: Social Media, Offensive, Trolling, Bullying, Abusive, Artificial Intelligence, Machine Learning, Detection, Anti Trolling, Tweets, Analysis

I. INTRODUCTION

For many people round the world social media sites are an integrated part of their lifestyle. There are many different social media sites supporting a good range of practices and interests. Social networks like Facebook, Twitter, Instagram and LinkedIn have become a source for news and a platform for political and moral debate for tons of users this is where trolling comes in, particularly, a troll often uses an

aggressive offensive language and has the aim to hamper the normal evolution of a web discussion and possibly to interrupt it. Only recently has it been possible to pay proper attention to the present problem, in order that many renowned press bodies and magazines have begun to address the difficulty and to write down articles both on the overall description of the phenomenon and on particular events that have caused a stir, favored by the increasing occurrence of behavior just like the one

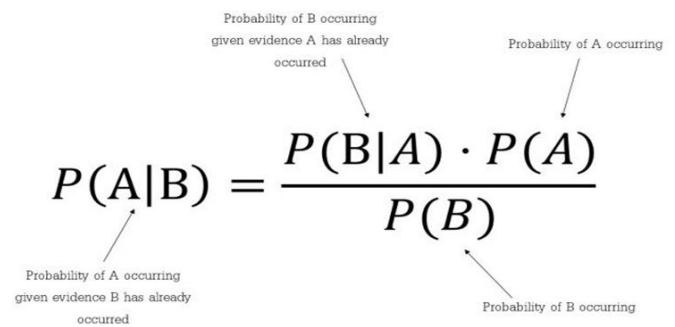
described above. Stories with different degrees of truthfulness accompanied by abusive language and trolling of either individual or a group are spread and tiny source criticism is applied by regular people also as journalists. Such an implementation would be interesting to the politicians, media, social networks or organizations that are targeted since it might be used to clear their name.

In this paper we discuss our system implemented by using Artificial Intelligence concept NLP - Natural Language Processing and Machine learning algorithms which Pre-process the data, Trains and tests the model, classifies the data into suitable categories of trolls and finally predicts and displays the result using visual representation tools like bar graph and pie chart.

II. ALGORITHMS AND TECHNIQUES

A. Naive Bayes

Naive Bayes is a simple technique for constructing classifiers: models that assign class labels to problem instances, represented as vectors of feature values, where the class labels are drawn from some finite set. There is not a single algorithm for training such classifiers, but a family of algorithms based on a common principle: all naive Bayes classifiers assume that the value of a particular feature is independent of the value of any other feature, given the class variable. For example, a fruit may be considered to be an apple if it is red, round, and about 10 cm in diameter. A naive Bayes classifier considers each of these features to contribute independently to the probability that this fruit is an apple, regardless of any possible correlations between the color, roundness, and diameter features. Abstractly, naïve Bayes is a conditional probability model: given a problem instance to be classified, represented by a vector.



$$P(A|B) = \frac{P(B|A) \cdot P(A)}{P(B)}$$

Fig. 1 - Naive Bayes Formula

B. Random Forest

Random forest is a supervised learning algorithm which is used for classification problems. Random forest algorithm creates decision trees on data samples and then gets the prediction from each of them and finally selects the best solution by majority of voting. While working with random forest in the first step, we create a bootstrap dataset out of the original dataset. Bootstrap dataset means shuffling of records, removal of duplicates and creating samples. In the second step, we prepare a decision tree from the bootstrap dataset. Prediction which holds the output is returned to the classifier shows the final answer as 0 or 1.

C. Working

The tweets collected need to be analyzed so as to assign labels. Using classifier labels are assigned to twitter data. Using Naive Bayes Technique tweets are classified either into a troll or not a troll. In Naïve Bayes, if a certain attribute is present then it is labelled as “1” or else it is “0”. By Naive-Bayes rule, probability of relevance for a document is calculated. It is assumed that attributes are not related to each other. For identification purposes, a feature is also labelled as an attribute. Classifying the tweets has various processes like collecting the tweets from twitter. Preprocessing the tweets, dividing the tweets and classifying by trainer. In dividing the tweets, the

training dataset is grouped into 5 different sets. While comparing, the validation part includes around 25 tweets. Grouping of selected tweets are done randomly. So these are some basic steps incorporated in this process. The NLTK library from python is used to carry out sentimental analysis. Naïve-Bayes algorithm classifies sentiments for remaining tweets. Previous trained data is implemented as input for this purpose.

Now to carry out sentimental analysis, Naïve Bayes classifier algorithm is used. Firstly, a training set consisting of positive words and negative words is created. The positive words are labelled as class “1” whereas the negative words are labelled as class “0”. This training set consists of 2005 positive words and 4783 negative words. New training sets can be made after scaling up this Dataset. The accuracy of the predicted labels is analyzed through performance parameters. The performance is represented in a form of matrix which is called confusion matrix. Confusion matrix is plotted to sum up the performance of the learning model. A confusion matrix for classes “P” and “N” can be represented as-

		Predicted class	
		P	N
Actual Class	P	True Positives (TP)	False Negatives (FN)
	N	False Positives (FP)	True Negatives (TN)

Here,

TP - The actual class as well as the predicted class is positive.

FN - The actual class is positive but the predicted class is negative.

FP - The actual class is negative but the predicted class is positive.

TN - The actual class as well as predicted class is negative.

Performance parameter are as follows:

Accuracy-

It replies to the question of “How often is the classifier correct?”

$$Accuracy = \frac{TrueNegatives + TruePositive}{TruePositive + FalsePositive + TrueNegative + FalseNegative}$$

In this paper using Naïve bayes technique the tweets are classified into eight categories i.e. Offensive, Not offensive, NULL, Individual, Group, Targeted, Untargeted and Others according to the trainer’s perception. This perception may vary with different dataset and situations

A specific amount of tweets are taken into consideration for this process and some keywords are selected from tweets for perception training. For example, 50 tweets are selected, then 40 tweets are trained and remaining is the test data. The results were verified by the trainer which were obtained by classification using the Naïve Bayes technique. Tweets collected are pre-processed and then given to naïve bayes classifiers. By training and verifying the sentiment classification by the same person, we could achieve a high degree of accuracy using Naïve Bayes technique. This method is suitable to train and classify sentiment from twitter and other social network data.

III. DATA AND METHODOLOGY

The dataset is given in *csv* file format with columns namely, ID, INSTANCE, SUBA, SUBB, SUBC where ID represents the identification number for the tweet, INSTANCE represents the tweets, SUBA consists of

the labels namely Offensive (OFF) and Not Offensive (NOT), SUBB consists of the labels namely Targeted Insult and Threats (TIN) and Untargeted (UNT) and SUBC consists of the labels namely Individual (IND), Group (GRP) and Other (OTH).

The dataset has 13240 tweets. All the instances are considered for Sub Task A. However, we have filtered and considered the data that are labelled with “TIN/UNT” and “IND/GRP/OTH” for Sub Task B and Sub Task C respectively by ignoring the instances labelled with “NULL”. Thus, we have obtained 4400 and 3876 instances for Sub Task B and Sub Task C respectively.

We have preprocessed the data by removing the URLs and the text “@USER” from the tweets. Tweet tokenizer 4 is used to obtain the vocabulary and features for the training data.

We have employed both traditional machine learning and deep learning approaches to identify the offensive language in social media. In deep learning (DL) approach, the tweets are vectorized using word embeddings and are fed into encoding and decoding processes. We have employed two attention mechanisms namely Normed Bahdanau (NB) and Scaled Luong (SL) in this approach. These two variations are implemented to predict the class labels for all the three sub tasks. These attention mechanisms help the model to capture the group of input words relevant to the target output label.

For example, consider the instance in Task C: “we do not watch any nfl games this guy can shove it in his pie hole”. This instance clearly contains the offensive slang “pie hole” and about watching the “nfl games”. The attention mechanism captures these named entities or groups of words and correctly maps to the label “GRP”.

In the traditional learning (TL) approach, the features are extracted from the tokens with a minimum count of two. The feature vectors are constructed using TF-IDF scores for the training instances. We have chosen the classifiers namely Multinomial Naive Bayes

(MNB) and Support Vector Machine (SVM) with Stochastic Gradient Descent optimizer to build the models for Task B and Task C respectively. These classifiers have been chosen based on the cross validation accuracies. The class labels namely “TIN/UNT” and “IND/GRP/OTH” are predicted for Task B and Task C using the respective models.

IV. IMPLEMENTATION

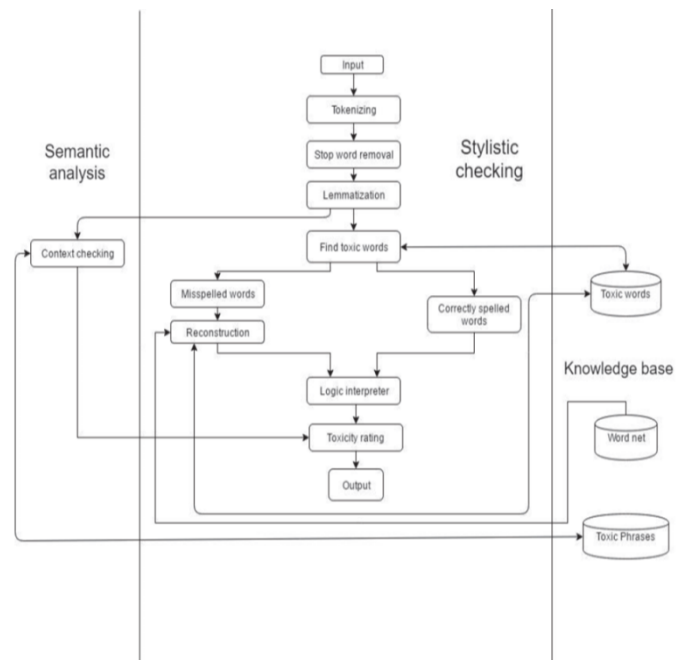


Fig.2 - System Architecture

The different phases of implementation and working of the system is discussed:

A. Semantic Analysis

In this layer, the contextual meaning of the sentence is going to be analyzed.

Context checking: The precise meaning of the sentence cannot be always understood by the literal meaning of the words utilized in the sentence. Hence during this part, the contextual meaning is taken into account.

B. Stylistic Checking

Input: The data used are tweets which is taken in csv file format and further processed in the following phases:

- 1) Tokenizing: The given sentence can't be easily understood by considering the whole sentence in one go. Hence, the sentence is weakened into the little part, i.e. one word per part referred to as a token. This manner helps in better understanding of the sentence.
- 2) Stop Word Removal: The words which don't contribute within the increase of toxicity of the sentence are mentioned as stop words. Such words (e.g. the, and, or) are deleted from the sentence during this step.
- 3) Lemmatization: The basic form of a word or its dictionary form is named lemma. Hence, during this part of the method, the basic form of the word is going to be returned, which can help in removing the inflectional endings and can make the method easier.
- 4) Stemming: It is the process of reducing a word to its word stem that affixes to suffixes and prefixes or to the roots of words known as a lemma. Stemming is important in natural language understanding (NLU) and natural language processing (NLP).
- 5) Data cleaning: It is a very crucial step in any machine learning model, but more so for NLP. Without the cleaning process, the dataset is often a cluster of words that the computer doesn't understand. Here, we will go beyond steps done in a typical machine learning text pipeline to clean data.
- 6) Words Recognition: The words which are recognized with the toxic words are categorized as offensive, targeted, individual, group etc and the words which don't contain any toxicity are categorized as not offensive or null.

- 7) Output: The output of the data received after being processed is accurate, reduced and free of any duplication making the further process easy.

C. Classification

- 1) Converting Words to Vector: Word Embeddings or Word vectorization is a methodology in NLP to map words or phrases from vocabulary to a corresponding vector of real numbers which is used to find word predictions, word similarities/semantics. The process of converting words into numbers are called Vectorization.
- 2) Loading and Labelling of Data: In NLP applications using Machine learning, loading the data is a crucial phase. As this data is loaded for 3 main purposes that is training the model, testing model, prediction. This is also where the data is labelled into different categories to make the model more effective.
- 3) Classification: It is also known as text tagging or text categorization is the process of categorizing text into organized groups. By using Natural Language Processing (NLP), text classifiers can automatically analyze text and then assign a set of pre-defined tags or categories based on its content.

D. Knowledge Base

This domain comprises all the databases that are required for training, testing and prediction stages of the system. It embodies toxic words and phrases classified in different categories which will be incorporated during training, testing and prediction phases.

V. RESULT AND EVALUATION

The performance is analyzed using the metrics namely precision, recall and training and testing accuracy.

The results of our approaches along with the Confusion matrix for our best run are presented in Tables 1, 2 and 3 for Task A, Task B and Task C respectively. We have obtained the best results for Task A MNB, Task B SVM, Task C RF models for Task A, Task B and Task C respectively.

```
Building Model Subtask A...
Preparing Test Data...
Training Accuracy: 1.0
Test Accuracy: 0.765625
Confusion Matrix:
[[243  0]
 [ 75  2]]
```

Table 1: Results of Confusion matrix for Sub-task A.

```
Building Model Subtask B...
Preparing Test Data...
Training Accuracy: 0.8484848484848485
Test Accuracy: 0.44155844155844154
Confusion Matrix:
[[34  0  0]
 [ 4  0  0]
 [39  0  0]]
```

Table 2: Results of Confusion matrix for Sub-task B.

```
Building Model Subtask C...
Preparing Test Data...
Training Accuracy: 1.0
Test Accuracy: 0.7647058823529411
Confusion Matrix:
[[ 0  4  0  0]
 [ 0 26  0  0]
 [ 0  3  0  0]
 [ 0  1  0  0]]
```

Table 3: Results of Confusion matrix for Sub-task C.

The attention mechanism Scaled Luong performs better when more data is available for training. The Normed Bahdanau attention mechanism performs better even for a small dataset. The deep learning model could not learn the features appropriately due to less domain knowledge imparted by the smaller data set. Thus, traditional learning performs better

with the given data size when compared to deep learning for Task C.

Results obtained are represented by using tools like bar graph and pie chart shown below:

Pie Chart for Offensive and Non-Offensive are presented in Fig 1 with 33.2% of Offensive data and 66.8% of Non-Offensive data

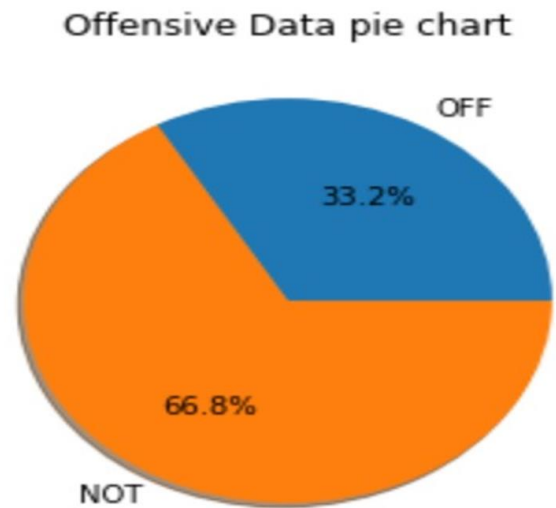


Fig 1: Pie Chart for Offensive and Non-Offensive.

The following Bar Graphs have Number of Tweets on Y-axis and Types of tweet on X-axis

Bar graph for sub task A in fig 2, based upon the offensive (OFF) and non offensive (NOT) data.

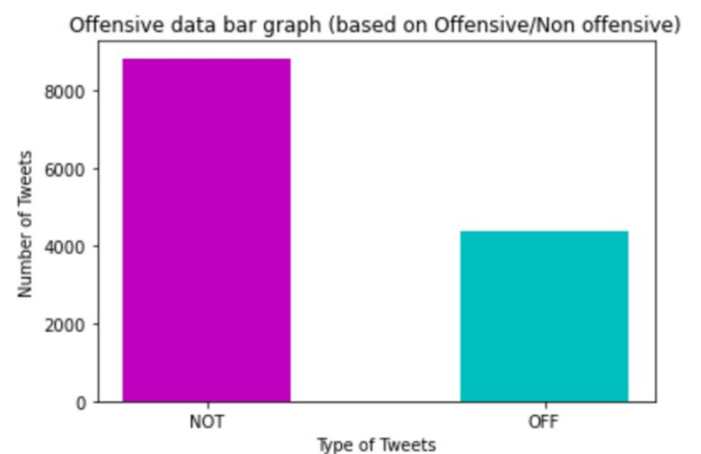


Fig 2: Bar Graph for Subtask A

Bar graph for sub task B in fig 3, based upon targeted (TIN) and untargeted data (UNT)

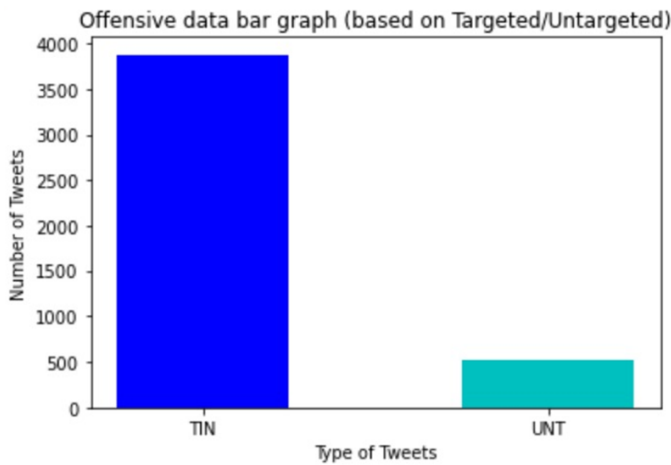


Fig 3: Bar Graph for Subtask B

Bar graph for sub task C in fig3, based upon individual (IND), group (GRP) and others (OTH).

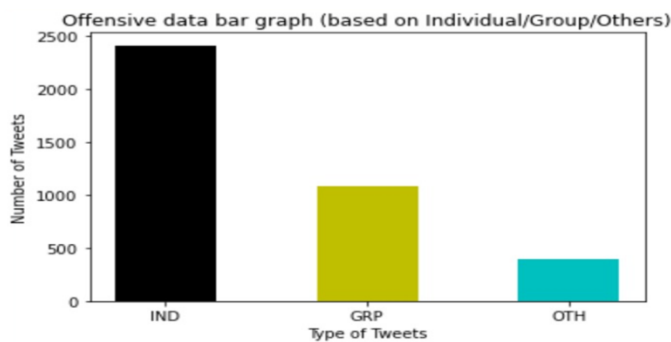


Fig 4: Bar Graph for Subtask C

VI. CONCLUSION

This paper has discussed the problems created by the presence of trolls in social media contexts and has presented the main approaches to tackle this problem. We have implemented a System using Artificial Intelligence and Machine learning, we have used different analysers like stylistic, syntactic etc this model uses Naïve bayes algorithm to get better accuracy in order to classify troll tweets. Our system provides various graphical analysis of the data, which helps users to identify the ratio of troll tweets and intensity of trolls.

The classifiers namely Multinomial, Naive Bayes and Support Vector Machine with Stochastic Gradient Descent optimizer were employed to build the models for the sub tasks. Deep learning with Scaled Luong attention, deep learning with Normed Bahdanau attention, and traditional machine learning with SVM give better results for Task A, Task B and Task C respectively. Our models outperform the baseline for all the three tasks.

VII. REFERENCES

- [1]. Zannettou, S.; Sirivianos, M.; Caulfield, T.; Stringhini, G.; De Cristofaro, E.; Blackburn, J. Disinformation warfare: Understanding state-sponsored trolls on twitter and their influence on the web. In Proceedings of the Web Conference 2019—Companion of the World Wide Web Conference, WWW 2019, San Francisco, CA, USA, 13–17 May 2019; pp. 218–226.
- [2]. Badawy, A.; Lerman, K.; Ferrara, E. Who falls for online political manipulation? In Proceedings of the Web Conference 2019—Companion of the World Wide Web Conference, San Francisco, CA, USA, 13–17 May 2019; pp. 162–168.
- [3]. Fornacciari, P.; Mordonini, M.; Poggi, A.; Sani, L.; Tomaiuolo, M. A holistic system for troll detection on Twitter. *Comput. Hum. Behav.* 2018, 89, 258–268.
- [4]. Donath, J.S. Identity and deception in the virtual community. In *Communities in Cyberspace*; Routledge: Abingdon-on-Thames, UK, 2002; pp. 37–68.
- [5]. Chun, S.A.; Holowczak, R.; Dharan, K.N.; Wang, R.; Basu, S.; Geller, J. Detecting political bias trolls in Twitter data. In Proceedings of the 15th International Conference on Web Information Systems and Technologies, WEBIST 2019, Vienna, Austria, 18–20 September 2019; pp. 334–342. [6] “<https://perspectiveapi.com/>”, last retrieved on 10th January 2017