VIVA-Tech International Journal for Research and Innovation
ISSN(Online): 2581-7280
Volume 1, Issue 4 (2021)
Article No. X
PP XX-XX

VIVA Institute of Technology
9th National Conference on Role of Engineers in Nation Building – 2021 (NCRENB-2021)

VIVA-TECH IJRI
ANNUAL RESEARCH JOURNAL

VIVA-TECH INTERNATIONAL JOURNAL FOR RESEARCH AND INNOVATION
ANNUAL RESEARCH JOURNAL
ISSN(ONLINE): 2581-7280

Categorize balanced dataset for troll detection

Prashant Singh[1], Nidhi Singh[2], Namit Rasalkar[3], Prof. Pallavi Raut[4]
[1](*Department of Computer Engineering, Mumbai University, India)*
Email: 15301015prashant@viva-technology.org
[2](*Department of Computer Engineering, Mumbai University, India)*
Email: 17301049nidhi@viva-technology.org
[3](*Department of Computer Engineering, Mumbai University, India)*
Email:17305024namit @viva-technology.org
[4](*Department of Computer Engineering, Mumbai University, India)*
Email: pallaviraut@viva-technology.org

Abstract : *As we know cyber bullying is increasing day by day and Cyber troll is one of the cyber-aggressive actions that is not much different from cyberbullying in online abuse so that the victims feel uncomfortable. One of the most used social media platforms in which cyber trolling frequently happens is Twitter. Basically, it is found that during an investigation of cyberbullying cases a lot of information gathered is false which aims to give discomfort, hatred and waste lots of time. So, it is necessary to classify between cyberbullying tweets and normal tweets on twitter. There has already been research on classification of cyberbullying tweets and normal tweets using the Support vector machine (SVM) algorithm. But the drawback of the system is that it only gives 63.83% of accuracy. Firstly, we can improve the accuracy of the system by using the Recurrent Neural Network (RNN) And Secondly, for balancing the dataset we will be using Synthetic Minority Over-sampling Technique (SMOTE). We believe that using these techniques we will be able to increase the accuracy of the previous proposed.*

Keywords- *Cyber bullying, Twitter, RNN, SVM, SMOTE*

I. INTRODUCTION

In this technical era, people are more connected with each other by means of the internet using many social networking sites. But these social networking sites can harm you sometimes by the means of cyberbullying, trolls, cyber threats etc. As we know Cyber bullying is increasing day by day and Cyber troll is one of the cyber-aggressive actions that is not much different from cyberbullying in online abuse so that the victims feel uncomfortable. One of the most used social media platforms in which cyber trolling frequently happens is Twitter. Basically, it is found that during an investigation of cyberbullying cases a lot of information gathered is false which aims to give discomfort, hatred and waste lots of time.

So, it is necessary to classify between cyberbullying tweets and normal tweets on twitter. To classify between cyberbullying tweets and normal tweets there are many machine learning algorithms available. Natural language processing will help machine learning better understand text and text analysis. Deep learning is one among the foremost advanced scientific fields of AI which will be applied here to yield far better results.

The system uses Recurrent neural network (RNN), which is a class of artificial neural networks where connections between nodes form a directed graph along a temporal sequence. This allows it to exhibit temporal dynamic behavior. Derived from feedforward neural networks, RNNs can use their internal state (memory) to process variable length sequences of inputs. This makes them applicable to tasks such as unsegmented, connected handwriting recognition or speech recognition.

Deep learning is a subset of machine learning where artificial neural networks, algorithms inspired by the human brain, learn from large amounts of data. The deep learning algorithms feature self-learning representations, they depend upon ANNs that mirror the way the brain computes information. During the training process, algorithms use unknown elements in the input distribution to extract features, group objects and discover useful data patterns. Much like training machines for self-learning, this occurs at multiple levels, using the algorithms to build the models.

II. RELATED WORK

Ahmed AL Marouf, Rasif Ajwad, Adnan Ferdous Ashraf, et.al. [1], They have traced the online behavior of the people who are suspected to be trolls to find the difference between trolls and normal users. For identifying trolls, they proposed a method of creating a profile of personality traits using user's textual data. Many psycholinguistic tools have been used such as LIWC, SlangNet, Sent WordNet, SentiStrength, Colloquial WordNet. Feature engineering based on LIWC has been performed on the post comments of the suspected trolls as well as victims. As slang is widely used to defame someone in the comments, SlangNet and Colloquial WordNet is used. For detection of troll's classification algorithm based on predictive models has been applied such as Multinomial Naive Bayes (MNB), Decision tree and sequential minimal optimization.

Wanda Athira Luqyana, Beryl Labique Ahmadie, Ahmad Afif Supianto, et.al.[2], It describes that research has been done previously using support vector machines but the dataset used for training and testing was unbalanced and that yielded inaccurate results. The author proposed a way to counter this by using the K-Nearest Neighbor algorithm to balance the unbalanced dataset. Previously SMOTE was used with SVM to balance the dataset but results could be improved with the new method. They used a dataset of 20,000 tweets for this method. They concluded that for training only 600 tweets were required and more tweets fed to the training model did not increase the accuracy of the machine. After the testing was done the accuracy was 68% which was better than SMOTE with SVM and normal SVM.

Patxi Galan Garcia, José Gaviria de la Puerta, Pablo Garcia Bringas, et.al.[3], Presented a methodology to detect and associate these fake profile accounts on Twitter to a real profile within the same network by analyzing the comments and posts of both the accounts and checking for the similarity in the pattern of way both accounts are operated. They also deployed this tactic in the working environment to detect and stop a cyberbullying situation in a real elementary school. The idea behind this approach is that every fake account is followed by the real account of the user behind the fake one. And there is a possibility to link a trolling account to a corresponding real account of the user who is behind the fake account, this is done by analyzing different features present in the profile, like their tweets and data characteristics, including text, using machine algorithms.

Ushma Bhatt, Divya iyyani, Keshin Jani, Ms. Swati Mali et.al.[4], Explored many such software systems "anti-trolling systems" and analyzed their methodologies and technological challenges. The "anti-trolling systems" proposed in this paper are one built by Facebook, Twitter, SMC4(Social Media C4) an app that is claimed to be the world's first anti-trolling software and also Perspective API an attempt by Google. All these systems were analyzed and the author found that Facebook stops trolls by identifying fake accounts by profile pictures and using facial recognition software on it. Facebook bans these accounts for posting comments which are not appropriate by allowing users to report them. Twitter created a twitter account imposter buster which finds the fake accounts by checking vitriolic tweets and updating the database of impersonator accounts. The system automatically replies to tweets of troll accounts and finds the evidence.

Todor Mihaylov, Preslav Nakov et.al. [5], Focuses on the trolls who manipulate people thus changing the public discourse in very important matters like in Bulgaria and Eastern Europe. Here the tolls who manipulate public opinion are divided into two categories: paid trolls that have been revealed by leaks used by "reputation management contracts" and "mentioned trolls" that have been reported such by several different people on social networks. Then the algorithms were trained on the data and tested with results such that it was 81-82% accurately predicted if it was a paid troll or a non-troll and same with accuracy it can predict if it was a mentioned troll or a non-troll. The authors collected a sample of 10,150 paid troll comments from the leaked reputation management documents, from Facebook and news community forums. For mentioned trolls they collected a total 1,140 comments that have been reported as trolls by the users of the platform. Features for distinguishing trolls from non-trolls were set by Bag of words, Bag of stems, Word n-grams, word prefix, word suffix, Emoticons, Punctuation count, Word2Vec clusters, Sentiment and Bad words. They trained and evaluated an L2-regularized Logistic Regression with LIBLINEAR as implemented in SCIKIT-LEARN, using scaled and normalized features to the [0;1] interval. For training data, the dataset was perfectly balanced of 650 negative and positive examples for paid trolls and non-trolls and 578 for mentioned trolls vs. non-trolls.

Amit Pratap Singh, Maitreyee Dutta, et.al.[6], To identify the spam accounts from non-spam accounts machine learning algorithm was deployed. For initial data was collected from H-Spam14 site and then different Preprocessing was applied to convert data which was easy for machine learning algorithm to understand, such as converting data into lowercase, then removal of stop word and after this the data will go through feature extraction phase, in which tokenization of words is done by dividing entire sentences into groups and extracted the best features from the raw data. To select the best features from the extracted feature set Artificial Bee Colony has been applied as an optimization algorithm to determine the optimal feature sets from spam and non-spam data sets. After the feature extraction classification process starts using the Artificial Neural Network to separate spam and non-spam data. They used Naive bayes and Support Vector Machines for separation of normal text from the

spam text. The accuracy of the system was about 99.14% by models which will learn the spam activities and maintain high accuracy for spam detection in the new tweets.
Dr. Priyanka Harjule, Astha Gurjar, Harshita Seth, Priya Thakur, et.al.[7], Have performed a Sentiment Analysis on the tweets that were published on twitter. The main focus of sentiment analysis is to classify the problem and predict the polarity of words as into positive or negative sentiment. Classifiers used in this experiment were mainly of two types, lexicon-based and machine learning based. The lexicon-based analysis included SentiWordNet and Word Sense Disambiguation while the machine learning based analysis included Multinomial Naive Bayes, Logistic Regression, Support Vector Machine (SVM) and RNN Classifier. Dataset was collected from an existing one from "Sentiment140" from Stanford University which contained 1.6 million tweets and the other dataset came from "Crowdflower's Data for Everyone" library totaling 13870 entries of tweets. Then MNB, LR, SVM Textblob, Sentiwordnet, and RNN classifiers were applied on the dataset that were obtained earlier and a comparison between the algorithm and its accuracy was done from the results obtained. While doing the machine learning approaches MNB, LR and SVM were assembled together to better classify the tweets and the models were ready to predict the sentiments of new data which will be fed to it. The authors found out that RNN among all the Machine learning algorithms performed better as it learns from the whole sentence unlike other algorithms which only checks the words as separate entities. SVM was better than MNB and LR as it does not ignore the interdependencies of the words in the sentence.

III. METHODOLOGY

The system uses Recurrent neural network (RNN), which is a class of artificial neural networks where connections between nodes form a directed graph along a temporal sequence. This allows it to exhibit temporal dynamic behavior. Derived from feedforward neural networks, RNNs can use their internal state (memory) to process variable length sequences of inputs. This makes them applicable to tasks such as unsegmented, connected handwriting recognition or speech recognition. In a RNN the information cycles through a loop. RNN classifies between cyberbullying tweets and normal tweets. For cleaning the data, Natural Linguistic Tools (NLP) is used. NLP is the sub-field of AI that is focused on enabling computers to understand and process human languages. For balancing of the dataset, the system will use Synthetic Minority Over-sampling Technique (SMOTE). The minority class is over-sampled by taking each minority class sample and introducing synthetic examples along the line segments joining any/all of the k minority class nearest neighbors. Depending upon the amount of over-sampling required, neighbors from the k nearest neighbors are randomly chosen.

The main algorithm for classification we are going to use is RNN. A recurrent neural network (RNN) is a class of artificial neural networks where connections between nodes form a directed graph along a temporal sequence. This allows it to exhibit temporal dynamic behavior. Derived from feedforward neural networks, RNNs can use their internal state (memory) to process variable length sequences of inputs. This makes them applicable to tasks such as unsegmented, connected handwriting recognition or speech recognition.

Fig 1, depicts First the dataset will be acquired from the Dataturks website and then preprocessing will be done. During the preprocessing phase the data will be examined. Since the dataset is text. it is necessary to go through the preprocessing stage so that it can be processed by the system. Pre-processing is the initial stage of text mining to extract knowledge. This process is done to dig, process and organize information and to analyze textual relationships from structured data and non-structural data. Applying Natural language processing During this stage case folding to convert all letter to upper or lower case, text cleaning to remove all unwanted characters, stop word removal to remove commonly used word (such as "the","a","an"), stemming to reduce inflected (or sometimes derived) word to their word stem, and tokenization to break up sequence of strings into pieces such as words.

Applying SMOTE after that data balancing is done using an oversampling technique so more data will be generated which is better for RNN classifiers to extract more features. One approach to addressing imbalanced datasets is to oversample the minority class. The simplest approach to balancing is done by duplicating examples in the minority class, although these examples don't add any new information to the model. Instead, new examples can be synthesized from the existing examples.

Text feature extraction plays a crucial role in text classification like tweets, influencing the accuracy of text classification. It is based on VSM, in which a text is viewed as a dot in N-dimensional space. Datum of each dimension of the dot represents one (digitized) feature of the text. And the text features usually use a keyword set. It means that on the basis of a group of predefined keywords, we compute weights of the words in the text by certain methods and then form a digital vector, which is the feature vector of the text. Text feature extraction methods which are currently used include filtration, fusion, mapping, and clustering method. After feature

extraction the data will be fed to RNN and weights to features will be applied in the neural network. And training and validation data will be separated for training purposes.

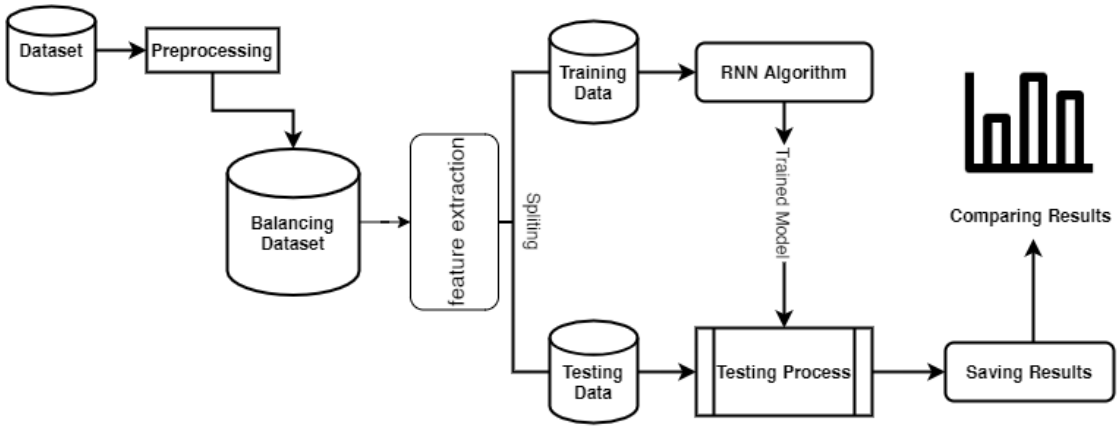

Figure 1: System Flow Diagram

IV. RESULT

```
In [8]: count_classes = pd.value_counts(df['label'], sort = True)
        count_classes.plot(kind = 'bar', rot=0)
        plt.xticks(range(2))
        plt.xlabel("label")
        plt.ylabel("frequency")

Out[8]: Text(0, 0.5, 'frequency')
```

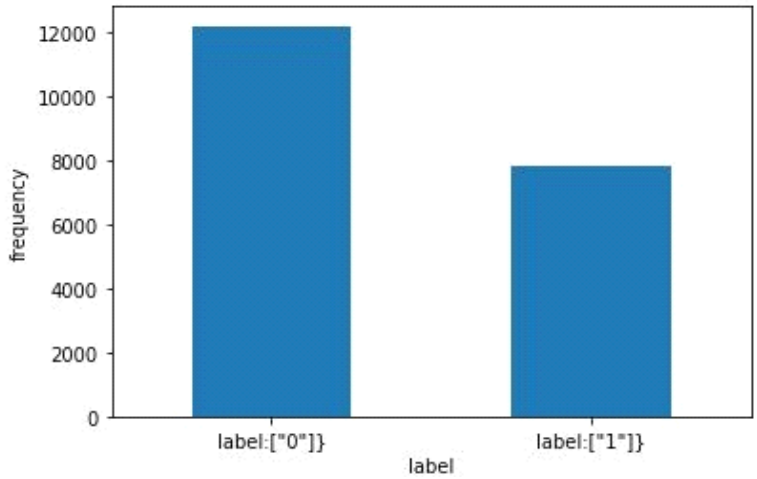

Fig 2. Unbalanced dataset

This is a brief information about partial implementation of the project and screenshots of the project Figure 3 shows that the dataset used is unbalanced. label [“0”] are non-troll tweets and label [“1”] are troll tweets and will be cleaned in the next procedure.

```
In [9]: stemmer = SnowballStemmer('english')
        words = stopwords.words("english")

In [10]: res['cleaned'] = res['content'].apply(lambda x:" ".join([stemmer.stem(i) for i in re.sub("[^a-zA-Z]", '

In [11]: res.tail()

Out[11]:
```

	content	extras	notes	label	cleaned
19996	I dont. But what is complaining about it goi...	NaN		[0]	i dont but complain go
19997	Bahah yeah i&;m totally just gonna&; get pis...	NaN		[0]	bahah yeah total gonna get piss talk mhm that ...
19998	hahahahaha >:) im evil mwahahahahahahahahaha	NaN		[0]	hahahahaha im evil mwahahahahahahahahaha
19999	What&;s something unique about Ohio? :)	NaN		[0]	what someth uniqu ohio
20000	Who is the biggest gossiper you know?	NaN		[0]	who biggest gossip know

Fig 3. Preprocessing

After the preprocessing techniques applied the data is very easy for the machine to understand and accuracy of the model will be increased due to this process.

```
from imblearn.over_sampling import SMOTE
smote = SMOTE()
```

```
X_train_smote, y_train_smote = smote.fit_sample(X_train, y_train)
```

```
C:\Users\P-Home\AppData\Roaming\Python\Python37\site-packages\sklearn\util
as keyword args. From version 0.25 passing these as positional arguments w
  FutureWarning)
```

```
from sklearn.preprocessing import MultiLabelBinarizer
y_train = MultiLabelBinarizer().fit_transform(y_train)
y_test = MultiLabelBinarizer().fit_transform(y_test)
X_train = MultiLabelBinarizer().fit_transform(X_train)
X_test = MultiLabelBinarizer().fit_transform(X_train)
```

```
from collections import Counter
print("Before Smote :", Counter(y_train))
print("After Smote :", Counter(y_train_smote))
```

```
con1 = np.count_nonzero(y_train_smote == 0)
con2 = np.count_nonzero(y_train_smote == 1)
print(con1,con2)
```

```
9709 9709
```

Fig 4. Balancing

V. CONCLUSION

Categorizing tweets into harmful trolls and non-trolls in a very efficient way is very helpful in finding the victim and culprit in a social setting. As per the literature survey various algorithms were used for identification of aggressive content, of identifying the psychology behind the troll tweet or to apply sentiment analysis on twitter content but we conclude that RNN is best served with natural language analysis and classification with much

better accuracy than Support vector machine which only gave 63.83% of accuracy from a dataset of 20,000 tweets. This will help moderating twitter content on creating a healthy environment for expressing free views.

Acknowledgements

We would like to express a deep sense of gratitude towards our guide Prof. Pallavi Raut, Computer Engineering Department for her constant encouragement and valuable suggestions. The work that we are able to present is possible because of his timely guidance. We would like to pay gratitude to the panel of examiners Prof. Sunita Naik, Prof. Dnyaneshwar Bhabad, & Prof. Vinit Raut for their time, effort they put to evaluate our work and their valuable suggestions time to time. We would like to thank Project Head of the Computer Engineering Department, Prof. Janhavi Sangoi for her support and coordination. We would like to thank the Head of the Computer Engineering Department, Prof. Ashwini Save for her support and coordination. We are also grateful to the teaching and the non-teaching staff of the Computer Engineering Department who lend their helping hands in providing continuous support.

REFERENCES

[1] Ahmed AL Marouf, Rasif Ajwad, Adnan Ferdous Ashraf, "Looking Behind the Mask: A framework for Detecting Character Assassination via Troll Comments on Social media using Psycholinguistic Tools.", International Conference on Electrical, Computer and Communication Engineering (ECCE), 2019.

[2] Wanda Athira Luqyana, Beryl Labique Ahmadie, Ahmad Afif Supianto, "K-Nearest Neighbors Undersampling as Balancing Data for Cyber Troll Detection", IEEE, 2019.

[3] Patxi Galan Garcia, José Gaviria de la Puerta, Pablo Garcia Bringas, "Supervised machine learning for the detection of troll profiles in twitter social network: application to a real case of cyberbullying", http://jigpal.oxfordjournals.org/, last accessed on: 3/11/2020.

[4] Ushma Bhatt, Divya iyyani, Keshin Jani, Ms. Swati Mali, "Troll-detection systems Limitations of troll detection systems and AI/ML anti-trolling solution", International Conference for Convergence in Technology - IEEE, 2018.

[5] Todor Mihaylov, Preslav Nakov, "Hunting for Troll Comments in News Community Forums", 54th Annual Meeting of the Association for Computational Linguistics, 2016.

[6] Amit Pratap Singh, Maitreyee Dutta, "Spam Detection in Social Networking Sites using Artificial Intelligence Technique", International Journal of Innovative Technology and Exploring Engineering (IJITEE), 2019.

[7] Dr. Priyanka Harjule, Astha Gurjar, Harshita Seth, Priya Thakur, "Text Classification on Twitter Data", IEEE, 2020.