International Journal of Scientific Research in Computer Science, Engineering and Information Technology
© 2017 IJSRCSEIT | Volume 2 | Issue 5 | ISSN : 2456-3307

To Discover Trolling Patterns in Social Media: Troll Filter

Pooja M. Tayade*[1], Shafi S. Shaikh[2], Dr. S. N. Deshmukh[3]

[1,2]Department of Computer Science and Information Technology, Dr. Babasaheb Ambedkar Marathwada University, Aurangabad, Maharashtra, India
[3]Department of Computer Science and Information Technology, Dr. Babasaheb Ambedkar Marathwada University, Aurangabad, Maharashtra, India

ABSTRACT

Troll - the word itself defines everything about bullying or harassing someone. Trolling is an important problem in the online world. Emotions or feelings perform a vital role in successful and effective communication between humans. However, this human communication is getting worst if there is a group of people who enjoy targeting someone and trolling, this happens in different social media like Facebook and Twitter. In this paper tried to filter out comments, which are negative or insulting. Goal of this paper is to identify the targets of trolls, so as to prevent trolling before it happens. For this purpose used sentiment analysis (Positive or Negative) through machine learning. The major focus of this paper was on comparing different machine learning algorithms for the task of sentiment classification. For classification, many classifiers are available but results are very promising with Naive Bayes, Support Vector Machines (SVM) and Maximum Entropy (MaxEnt) classifiers. The major findings were evaluated that the Support Vector classifier provides the highest classification accuracy for this domain.

Keywords: Sentiment Analysis, Unigram – Bigram dependencies, Training data, Test data, Tweets, classifiers-SVM, Naïve-Bayes, MaxEntropy

I. INTRODUCTION

Sentiment Analysis is a Natural Language Processing and Information Extraction task that aims to obtain collecting and analyzing data based upon the person feelings, reviews and thoughts. Sentimental Analysis is done by using various machine learning techniques, statistical models and Natural Language Processing (NLP) for the extraction of feature from a large data [1]. NLP works on the basis of two methods one is to build dictionary depending on previous words or use ML algorithms.

Twitter is a micro blogging platform where anyone can read or write short form of message which is called tweets. The amount of data accumulated on twitter is very huge. This data is unstructured and written in natural language. Twitter Sentimental Analysis is the process of obtaining tweets for a particular topic and predicts the sentiment of these tweets as positive or negative with the help of different machine learning algorithms.

In this paper the goal is to identify the negative comments. Mainly focusing on identifying the possible targets of trolls [2]. A troll is someone who creates quarrel on sites like Facebook and Twitter by posting messages that are particularly disputable or provoking with the sole intent of inflammatory an emotional response from other users. These type of messages are habitually distracting and take focus away from the subject at hand, sending a judicious communication down a rabbit hole of rudeness, personal attacks and jokes about your parents.

Trolls are malignant users who post or spread misleading, offensive or nonsensical information on the network. Anaconda (Python3.6) version was used as it is a mature. Python is simple yet powerful, high-level, versatile, robust, interpreted and dynamic programming language, which is well known for its functionality of processing natural language data by using NLTK

(Natural Language Toolkit) which is used for this paper. It is an interpreted language which makes the testing and debugging extremely quickly as there is no compilation step. There are extensive open source libraries available for this version of python and a large community of users.

NLTK is a library of python, which provides a base for building programs and classification of data. NLTK also provide graphical demonstration for representing various results or trends and it also provide sample data to train and test for various classifiers respectively. NLTK has enriched with all set of languages and is growing day by day.

II. RELATED WORK

Research work in the area of Sentiment analysis is numerous. There are some works done by authors on trolling in different ways:

Jorge et al. [3] worked on filtering troll comments using method called as comprehension models to filter out trolling comments. They used this approach with data from 'Meneame', a popular Spanish social news sites. They categories the comments in three different classifications and they have several possible classes as follows:

- Type of Information
- Focus of the comment
- Controversy Level

They used different methodologies like Cross validation, Learning the model and testing the models. For classification they used SVM and compression algorithm while SVM results in 76.56% accuracy whereas compression algorithm gives 75.74% accuracy.

Paraskevas et al. [4] worked on troll vulnerability and they performed Troll Vulnerability Prediction algorithm. They predicting that, whether troll is vulnerable or not. For experiments they used Reddit recalling large fraction of the troll-vulnerable posts. They used 9541 comments as trolling and after classification 3853 comments were found vulnerable, which counts to about 2.5 trollings per vulnerable comment, on average.

Kumar et al. [5] showed pattern of trolls in Slashdot Zoo via decluttering. They developed a general algorithm called TIA (Troll Identification Algorithm) to classify users of an online 'Signed' social network as malicious (troll) or benign (honest users). They have shown that they can significantly improve on past works in the detection of trolls on Slashdot Zoo using a suite of decluttering operations that simplify a signed social network by removing confusing or irrelevant edges from the network. They proposed the TIA algorithm that takes any centrality measure and any set of decluttering operations as input parameters, and uses them to iteratively identify the trolls in the social network. Its running time is faster than many past algorithm and gives accurate result than existing methods. TIA using Signed Eigenvector Centrality and decluttering operations a and e gives the best result of 51.04%, significantly exceeding the 15.07% when no decluttering operations are performed.

When trying to seek work carried out by researchers in the concerned area, it seems that not much work is done to identify trolling nature. Some research work is ongoing and some techniques are also introduced for same purpose as discussed above but this amount of work is not sufficient. One technique that can be used to identify trolling nature in social media is try to understand the nature of spam mails or false mails in inbox. This work will give some hint of how to deal with unwanted data.

III. METHODOLOGY

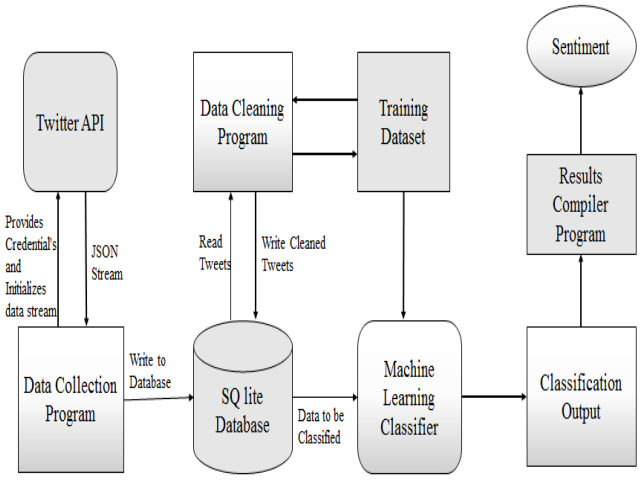

Figure 1. Architecture of Troll Filtering System [6]

A. Data Collection

To use Twitter API we must first have a twitter account. It can be easily created by filling the sign up details in

twitter.com website. After this you will be provided with a username and password which is use for login purpose. Once your account is created, you can now read and send tweets on any topic you want to explore. Twitter provider a platform from which we can access data from twitter account and can use it for our own purpose. For this we have to login with our twitter credentials in apps.twitter.com website. In this website, we first create an application which will be used for streaming tweets by providing necessary details. Once our API is created we can get to know customer key, customer secret key, access token key and access secret key. These keys are used to authenticate user when user want to access twitter data. We use all the keys and secrets which we got in API, we create our own dataset for positive and negative tweets through twitter API.

TABLE I. DATA COLLECTION FOR TOLL FILTER

Type	Tweets
Positive	10104
Negative	16175

B. Adjective-Noun Pair(ANP) Construction

Data collection process is carried out by collecting tweets through twitter API and created own database, there was a major focus on collecting negative tweets or comments. Troll data having some examples like Alia Bhatt and Donald Trump.

Ex.1) #AliaBhatt is so #dumb she thought #PaniPuri , #SevPuri #cholapuri are all relative of #amrishpuri & #ompuri @aliaa08 #aliabhatttroll #troll

Ex.2) No president has ever been more vulgar or more despicable than Donald Trump. #BrainlessTrump.

These examples show that they both were trolled by someone. Alia Bhatt was trolled by her GK and Donald trump for his tweets, speech or decisions. This way anyone can get troll on social media. This paper tries to work out troll filter goal in to two parts. First is, try to identify the comments are positive or negative. For this negative words and their combination with other words plays important role in discover negativity. Second is, try to filter out comments from user profile which cause troll. There was always the focus on negative polarity, as trolling is bullying someone it needs to be analyzed that we have to focus on negativity mostly. Overall 75% work will focus on Negative polarity of text. When trying to build troll filter word dictionary for troll filtering system the best way is to come across Adjective-Noun Pairs (ANPs). A single word cannot find the emotion of whole sentence. For example, 'Dumb' is the word which only gives information or just express emotion, someone has. But 'Dumb Alia' will justify that she is dumb and someone is trying to troll her. For this purpose we are using ANPs.

It is very important that a single noun and a single adjective cannot be focus on the negativity of the text. That's why we are using both Adjective and Nouns in combine. It is called as Adjective Noun Pairs. Examples of ANP are as Follows:

TABLE II. EXAMPLES OF POSITIVE AND NEGATIVE ANPS

Strictly Positive Pairs	Strictly Negative Pairs
Ambitious Person	Corrupt System
Adorable Place	Criminal Mind
Benevolence Guy	Cruel Husband
Amicable Teacher	Distrust Employee
Beautiful Girl	Greedy Dog
Sporty Girl	Horrible Face
Delicious Food	Inactive Mind
Smiley Face	Mischievous Bull
Zakkas Personality	Oblivious Donkey
Great Nature	Rubbish Decision
Good Actor	Scary Hand
Cracking Stunt	Glutton Boy

C. Data Pre-processing

1) Cleaning

Data obtained from twitter is not exactly fit for extracting features. Generally the tweets are like raw data which consists of message along with its usernames, empty spaces, special characters, emoticons, abbreviations, hash tags, time stamps, URL's ,etc. We pre-process on tweets by using various function of NLTK. In preprocessing we first extract our main message from the tweet, then we remove all empty spaces, hash tags, repeating words, URL's, etc. We also remove the repetition of tweets.

2) Tokenization

Tokenization is nothing but Segmentation of sentences. In this step, we will tokenize or segment text with the help of splitting text by spaces and punctuation marks to form container of words.

3) Removal of Stopwords

In the phase of stopword , articles such as ‘a’, ‘an’, ‘the’ and other stopwords such as ‘to’, ‘of’, ‘is’, ‘are’, ‘this’, ‘for’ removed.

TABLE III. OUTPUT AFTER CLEANING ORIGINAL DATA

Original Data	Data After Cleaning
2017-06-06 10:23:04,"b""@RealDonad_ Trump A legitimate question,'sir', Are you an Isis sympathizer? As you promote fear and hatred it seems so. #terror #Criminal"""	A legitimate question sir Are you an Isis sympathizer As you promote fear and hatred it seems so terror Criminal
2017-06-06 08:15:45,"b""#Fortune favors the #Brave.#FearlessFocus #LuxuryFitnessConsulting #Courage #Confident\xe2\x80\xa6 https://t.co/8aA32Jc6Kh"""	Fortune favors the Brave FearlessFocus LuxuryFitnessConsulting Courage Confident

TABLE IV. OUTPUT SHOWING TOKENIZED WORDS IN COMMENT

Before Tokenization	Data After Tokenization
A legitimate question sir Are you an Isis sympathizer As you promote fear and hatred it seems so terror Criminal	[“A”, ”legitimate”, ”question”, ”sir”, ”Are”, “you”, “an”, “Isis”, “sympathizer”, “As”, “you”, “promote”, “fear”, “and”, “hatred”, “it”, “seems”, “so”, “terror”, “Criminal”]
Fortune favors the Brave FearlessFocus LuxuryFitnessConsulting Courage Confident	[“Fortune”, “favors”, “the”, “Brave”, “FearlessFocus”, “LuxuryFitnessConsulting”, “Courage Confident”]

TABLE V. OUTPUT AFTER REMOVAL OF STOPWORDS FROM DATA

Before Removal of Stopwords	After Removal of Stopwords
[“A”, ”legitimate”, ”question”, ”sir”, ”Are”, “you”, “an”, “Isis”, “sympathizer”, “As”, “you”, “promote”, “fear”, “and”, “hatred”, “it”, “seems”, “so”, “terror”, “Criminal”]	[”legitimate”, ”question”, “Isis”, “sympathizer”, “promote”, “fear”, “hatred”, “seems”, “terror”, “Criminal”]
[“Fortune”, “favors”, “the”, “Brave”, “FearlessFocus”, “LuxuryFitnessConsulting”, “Courage Confident”]	[“Fortune”, “favors”, “Brave”, “FearlessFocus”, “LuxuryFitnessConsulting” , “Courage Confident”]

4) POS Tagging

It gives the more information about the tweets. There are many parts of speech such as noun, pronoun, adjective, verb, adverb, preposition, and so on. A word can take different meanings in different sentences, i.e. a word can act as a noun in one sentence, and as an adjective in another.

D. Post-Processing

After completion of pre-processing step next step is post-processing.

1) Dataset

On raw data we performed various processing techniques. Now, we are having proper collection of tweets which are positive as well as negative. We are having more than 25,000 collection of tweets, from those some are positive and others negative, which we called dataset. From collection of tweets ¾ part of data have taken as training data for both positive and negative and ¼ as a test data. So the total training data consist of 19709 tweets (in that 12131 tweets are negative whereas 7578 tweets are positive tweets). In case of testing data there are total 6570 tweets are considered for testing of trained classifier.

2) Classifiers

There are different types of machine learning algorithms for better performance and accuracy. In this paper we have used Naïve Bayes, SVM and MaxEnt for perfect accuracy and for comparison purpose. SVM gives better results than other classifiers.

2.1 Naïve Bayes

It is a simple classifier based on Bayes theorem and makes naive independence assumptions of the feature variables. It gives polarity between 0’s and 1’s with an assumption of independence among predictors.

```
from nltk.classify import NaiveBayesClassifier
```

For classification we must need to use the nltk function through this we import Naïve Bayes Classifier.

```
classifierName = 'Naive Bayes'
classifier = NaiveBayesClassifier.train(trainfeats)
```

2.2 SVM (Support Vector Machine)

Support Vector Machine is another popular classification technique. A support vector machine constructs a hyperplane or set of hyperplanes in a high-dimensional space such that the separation is maximum. This is the reason the SVM is also called the maximum margin classifier. The hyperplane identifies certain examples close to the plane which are called as support vectors. LinearSVC from sci-kit learn, which is a python package, is used to classify the tweets.

```
from sklearn.svm import LinearSVC
from nltk.classify import SklearnClassifier
```

```
classifierName = 'SVM'
classifier = SklearnClassifier(LinearSVC(), sparse=False)
classifier.train(trainfeats)
```

2.3 Maximum Entropy

The Max Entropy classifier is a discriminative classifier commonly used in Natural Language Processing, Speech and Information Retrieval problems. The MaxEnt classifier is based on the principle of maximum entropy and from all the models, chooses the once which has the maximum entropy. The goal is to classify the text (tweet, document, reviews) to a particular class, given unigrams, bigrams or others as features.

```
from nltk.classify import MaxentClassifier
```

```
classifierName = 'Maximum Entropy'
classifier = MaxentClassifier.train(trainfeats, 'GIS', trace=0,
encoding=None,labels=None,sparse=True,gaussian_
prior_sigma=0, max_iter =1)
```

IV. FEATURE EXTRACTION

A. Unigram

Unigrams are the simplest features that can be used for learning tweets. The bag-of-words model is a powerful technique in sentiment analysis. This technique involves collecting all words in the document and using them as features. The features can either be the frequency of words, or simply 0s and 1s to indicate if the word is present in the document or not. In this project, 0s and 1s are used to indicate the absence or presence of a word in the tweet [7].

B. Bigram

Bigrams are features consisting of sets of two adjacent words in a sentence. Unigram sometimes cannot capture phrases and multi-word expressions, effectively disregarding any word order dependence. For example, words like ‘brainless trump’, ‘hopeless people’ clearly say that the sentiment is negative, but a unigram might fail to identify this. In such cases, bigrams help in recognizing the correct sentiment of the tweet [7]. ANPs are also special type of Bigrams where multiword combination is made on certain conditions as Noun + Adjectives (brainless + Trump) or Noun + Adverb may define force of target to classify efficiently.

V. RESULT

For troll filtering it is very important to select best features from given dataset and apply supreme classification algorithm which gives better result as comment is positive or negative. In this paper, we tried to select unigram and bigram as features and NB, SVM and MaxEnt as algorithms. Navie bayes algorithm gives result with accuracy of 82% for unigram and 83% for bigram. Whereas 80% of accuracy for unigram and 81% for bigram features given by maximum entropy algorithm. While implementing all these algorithms we can conclude that SVM is the best algorithm for classification of comments with accuracy of 95% for unigram and 97% for bigram.

TABLE VI. EVALUATION OF DIFFERENT CLASSIFICATION MODELS FOR UNIGRAM FEATURES

Algorithm	Accuracy	Precision	Recall	F-measure
Naïve Bayes	83.411	83.876	85.736	83.253
MaxEnt	80.276	79.911	81.556	79.922
SVM	94.688	94.174	94.712	94.424

TABLE VII. EVALUATION OF DIFFERENT CLASSIFICATION MODELS FOR BIGRAM FEATURES

Algorithm	Accuracy	Precision	Recall	F-measure
Naïve Bayes	82.909	83.508	85.321	82.762
MaxEnt	80.687	80.395	82.083	80.360
SVM	97.415	97.660	96.901	97.253

VI. CONCLUSION

Understanding the trolling behaviour and trolling nature in social media platform is very essential and has attracted unavoidable and considerable attention towards it as trolling will affect personal relations among virtual as well as real society. The problem of detecting trolls in online environments like Twitter, Facebook, Instagram, Flicker and other signed social networks is increasingly important as open source, collaboratively edited information like comments, tweets and re-tweets used more widely and openly. We have carried out many classification algorithms to understand the concept of troll vulnerability to show how suspicious a post, comment or tweet is to troll. Then there is next step to filter out the troll by evaluating how much the troll is going to harm according to particular user.

Our initial results using the Twitter dataset are promising, suggesting that a proactive treatment of filtering out trolls is possible. The proposed system is, built and evaluated a detection system intend to aid social media user in automatically detecting negative comments of malicious, abusive and insulting intent. This methodology is not intended to automatically remove inappropriate content, as different sites have their own regulation. At this stage methodology only tries to detect troll content and inform the user that this comment is abusive so next step depends on user whether user wants to block or delete comment. SVM with promising accuracy for both feature selection showed much better result in the mode of deciding whether a comment or tweet has a troll in nature or not. In future work will focus on how to make it real time application and how to increase accuracy with better classifying word of dictionary.

VII. REFERENCES

[1] Prateek Garg ,"Sentiment Analysis of Twitter Data using NLTK in Python" ,computer science and engineering department, Thapar university, Patiala, june 2016.

[2] Erik Cambria, "Affective Computing and Sentiment Analysis", Published by the IEEE Computer Society, 2016.

[3] Jorge de-la-Pena-Sordo, Igor Santos, and Pablo G. Bringas, "Using Compression Models for Filtering Troll Comments" in IEEE 10th Conference on Industrial Electronics and Applications (ICIEA), June 2015.

[4] Paraskevas Tsantarliotis, Evaggelia Pitoura and Panayiotis Tsaparas, "Troll Vulnerability in Online Social Networks" in IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM), 2016.

[5] Srijan Kumar, Francesca Spezzano, and V.S. Subrahmanian, "Accurately Detecting Trolls in Slashdot Zoo via Decluttering" in IEEE, 2014. (conference style)

[6] John Dodd, "Twitter Sentiment Analysis" National college of Ireland, May 2014.

[7] Shachi H Kumar," Twitter Sentiment Analysis", University of California Santa Cruz Computer Science.