International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 08 Issue: 04 | Apr 2021 www.irjet.net p-ISSN: 2395-0072

MONITORING OF SUSPICIOUS DISCUSSIONS ON ONLINE FORUMS

Atharva Gadiwan[1], Anish Kumar[2], Janhavi Ghuge[3], Prof.Santosh Tamboli[4]

[1-3] *Student, Department of IT Engineering, Vidyalankar Institute of Technology, Mumbai, India*

[4] *Prof. Santosh Tamboli, Departmentof IT Engineering, Vidyalankar Institute of Technology, Mumbai, India*

Abstract - *Due to the substantial growth of internet users and its spontaneous access via electronic devices,the amount of electronic contents has been growing enormously in recent years through instant messaging, social networking posts, blogs, online portals and other digital platforms. Unfortunately, the misapplication of technologies has increased with this rapid growth of online content, which leads to the rise in suspicious activities. People misuse the web media to disseminate malicious activity, perform the illegal movement, abuse other people, and publicize suspicious contents on the web. The suspicious contents usually available in the form of text, audio, or video, whereas text contents have been used in most of the cases to perform suspicious activities. Thus, one of the most challenging issues for NLP researchers is to develop a system that can identify suspicious text efficiently from the specific contents. In this paper, a Machine Learning (ML)-based classification model is proposed to classify English text into non-suspicious and suspicious categories based on its original contents.*

Key Words: Illegal Activities, Discussion forums, Sentimental Analysis, Machine Learning.

1. INTRODUCTION

Accelerating crimes on digital mediums alert the law implementation bodies to continuously monitor online activities. To achieve the above we need to build a system which detects suspicious postings on online forums. A lot of surveys and facts have proved that it is difficult to manage information which constantly keeps changing on internet thus data mining is the optimal choice to analyse and gather data. Using various data mining techniques, raw data is extracted from a large text corpus and this raw /unstructured data is transformed into structured data in pre-processing. This paper highlights the datamining techniques and sentimental algorithm which is prototyped and implemented using python which is functional in natural language using Natural Language Toolkit (NLTK) library.

2. LITERATURE SURVEY

A research paper published [1], suggests various techniques and algorithms which can be employed. The paper elaborates about Stop-word Selection, Stemming algorithm, Brute-force algorithm, Learning Based algorithm and Matching algorithm. Matching algorithms use two constraints Stemmer Strength and Index Compression. Using these two constraints, stem words in database are compared and their value is calculated. Learning based algorithms include machine learning theories like SVM and conditional random field.

Another paper [2], describes the system will analyse data from few discussion forums and will classify the data into different groups i.e. legal and illegal data using Levenshtein algorithm. Levenshtein is used to measure similarity between two words.

In this paper work, they have used Social Graph generation based approach for the identification of suspicious users and chat logs. Overall process of graph based suspicious activity detection is performed in seven steps. These steps are Generation of instant chat application, Storage of user chat logs, Data extraction from chat logs, Data pre-processing & normalization, Key Information Extraction, Social Graph Generation, Suspicious Group Identification. By using these steps, suspicious activity can be identified. Here, Concept of SVM approach is used for the extraction of key information like key users, key terms and key sessions.

2. DETAILED WORK

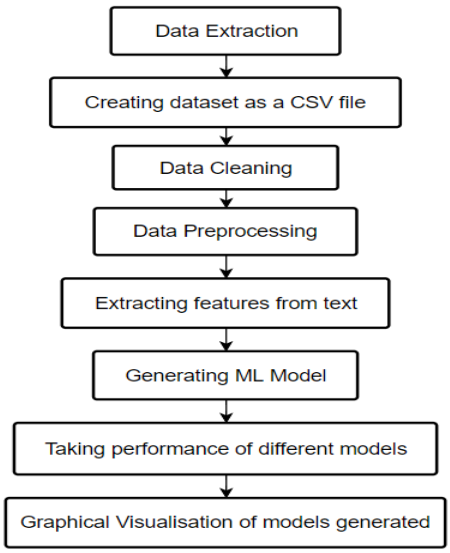

Fig-1: Architecture of System

Reddit is basically a large group of forums in which registered users can talk about almost anything you can imagine, from news, to pop culture, to technology, to comics, to film, to literature, to the weirdest things in the world, including Not Safe For Work stuff. PRAW is a Python wrapper for the Reddit API, which enables to scrape data from subreddits. The extracted data from reddit is saved into a CSV file.

3.1 Data Cleaning

Data Cleaning is important to identify and remove errors & duplicate data, in order to create a reliable dataset. This improves the quality of the training data for analytics and enables accurate decision-making. Beautiful soup is a python library for pulling data out of html and xml files. It works with your favorite parser to provide idiomatic ways of navigating, searching, and modifying the parse tree. The web generates tons of text data and this text might have HTML tags in it. These HTML tags do not add any value to text data and only enable proper browser rendering so they should be removed. Similarly, punctuation and numeric values don't add value to the text so they are removed.

3.2 Data Preprocessing

Data preprocessing is an important task. It is a data mining technique that transforms raw data into a more understandable, useful and efficient format.

3.2.1 Tokenization

Given a character sequence and a defined document unit, tokenization is the task of chopping it up into pieces, called tokens, perhaps at the same time throwing away certain characters, such as punctuation. Here is an example of tokenization:

Input: "It originated from the idea that there are readers who prefer learning new skills from the comforts of their drawing rooms"

Output:['It', 'originated', 'from', 'the', 'idea', 'that', 'there', 'are', 'readers', 'who', 'prefer', 'learning', 'new', 'skills', 'from', 'the', 'comforts', 'of', 'their', 'drawing', 'rooms']

3.2.2 Stop words selection

Stop words are the most used words in the English language which includes the words pronouns such as “I, he, she” or articles such as “a, an, the” or prepositions. Information Retrieval (IR) systems was first introduced the concept of stop-words . For a significant portion of the text size in terms of frequency of appearance small portion of words in the English language accounted. It was noticed that the mentioned pronouns and preposition words were not used as index word to retrieve documents. Thus, it was concluded that such words did not carry significant information about documents. Thus, the same interpretation was given stop words in text mining applications as well .To reducing the size of the feature space the standard practice of removing stop words from the feature space is mainly used. The stop word list that is considered to be removed from the feature space generic stop words list which is application independent . This may have an adverse effect on the text mining application as certain word is dependent on the domain and the application

3.2.3 Lemmatization

A word in a text may exist in multiple forms like stop and stopped (past participle or price and prices (plural). Text normalization converts variations of the word into root form of the same word.

Lemmatization does not simply chop off inflections, but instead relies on a lexical knowledge base like WordNet to obtain the correct base forms of words

Table -1: Lemmatization

Original word	*Lemmatized word*
connected	connect
trouble	trouble
consulting	consult
studies	study
walked	walk

3.3 Feature extraction using TF-IDF Vectorizer

TF-IDF is a combination of two different words i.e. Term Frequency and Inverse Document Frequency. First, the term “term frequency” will be discussed. TF is used to measure that how many times a term is present in a document.

•*Term Frequency [TF]*

Let's suppose, we have a document "T1" containing 5000 words and the word "Alpha" is present in the document exactly 10 times. It is very well known fact that, the total length of documents can vary from very small to large, so it is a possibility that any term may occur more frequently in large documents in comparison to small documents. So, to rectify this issue, the occurrence of any term in a document is divided by the total terms present in that document, to find the term frequency. So, in this case the term frequency of the word

"Alpha" in the document "T1" will be

TF = 10/5000 = 0.002

•*Inverse Document Frequency (IDF)*

When the term frequency of a document is calculated, it can be observed that the algorithm treats all keywords equally, doesn't matter if it is a stop word like "of", which is incorrect.

All keywords have different importance. Let's say, the stop word "of" is present in a document 2000 times but it is of no use or has a very less significance, that is exactly what IDF is for. The inverse document frequency assigns lower weight to frequent words and assigns greater weight for the words that are infrequent. For example, we have 10 documents and the term "technology" is present in 5 of those documents, so the inverse document frequency can be calculated as [4]

IDF = log_e (10/5) = 0.3010

•*Term Frequency - Inverse Document Frequency (TF-IDF)*

Now it is understood that, the greater or higher occurrence of a word in documents will give higher term frequency and the less occurrence of word in documents will yield higher importance (IDF) for that keyword searched in particular document. TF-IDF is nothing, but just the multiplication of term frequency (TF) and inverse document frequency (IDF). We have already calculated TF and IDF. To calculate the TF-IDF:

TF-IDF = 0.002*0.3010 = 0.000602

3.4 Machine Learning model generation

- *Bayes' Theorem*

This lets us examine the probability of an event based on the prior knowledge of any event that related to the former event. So for example, the probability that price of a house is high, can be better assessed if we know the facilities around it, compared to the assessment made without the knowledge of location of the house. Bayes' theorem does exactly that.

$$P(A \mid B) = \frac{P(B \mid A)\, P(A)}{P(B)}$$

Above equation gives the basic representation of the Bayes' theorem. Here A and B are two events and,

P(A|B) : the conditional probability that event A occurs , given that B has occurred. This is also known as the posterior probability.

P(A) and P(B) : probability of A and B without regard of each other.

P(B|A) : the conditional probability that event B occurs , given that A has occurred.

Naive Bayes Algorithm- The complexity of the above Bayesian classifier needs to be reduced, for it to be practical. The naive Bayes algorithm does that by making an assumption of conditional independence over the training dataset. This drastically reduces the complexity of above mentioned problem to just 2n.

The assumption of conditional independence states that, given random variables X, Y and Z, we say X is conditionally independent of Y given Z, if and only if the probability distribution governing X is independent of the value of Y given Z.

In other words, X and Y are conditionally independent given Z if and only if, given knowledge that Z occurs, knowledge of whether X occurs provides no information on the likelihood of Y occurring, and knowledge of whether Y occurs provides no information on the likelihood of X occurring.

This assumption makes the Bayes algorithm, naive.

Given, n different attribute values, the likelihood now can be written as

$$P(X_1 \ldots X_n|Y) = \prod_{i=1}^{n} P(X_i|Y)$$

Here, X represents the attributes or features, and Y is the response variable. Now, P(X|Y) becomes equal to the products of, probability distribution of each attribute X given Y.

- *SVM Algorithm*

The SVM algorithm is implemented in practice using a kernel. The learning of the hyperplane in linear SVM is done by transforming the problem using some linear algebra, which is out of the scope of this introduction to SVM.

A powerful insight is that the linear SVM can be rephrased using the inner product of any two given observations, rather than the observations themselves. The inner product between two vectors is the sum of the multiplication of each pair of input values.

For example, the inner product of the vectors [2, 3] and [5, 6] is 2*5 + 3*6 or 28.

The equation for making a prediction for a new input using the dot product between the input (x) and each support vector (xi) is calculated as follows:

f(x) = B0 + sum(ai * (x,xi))

This is an equation that involves calculating the inner products of a new input vector (x) with all support vectors in training data. The coefficients B0 and ai (for each input) must be estimated from the training data by the learning algorithm.

- *SGD Algorithm*

The word 'stochastic' means a system or a process that is linked with a random probability. Hence, in Stochastic Gradient Descent, a few samples are selected randomly instead of the whole data set for each iteration. In Gradient Descent, there is a term called "batch" which denotes the total number of samples from a dataset that is used for calculating the gradient for each iteration. In typical Gradient Descent optimization, like Batch Gradient Descent, the batch is taken to be the whole dataset. Although, using the whole dataset is really useful for getting to the minima in a less noisy and less random manner, but the problem arises when our datasets gets big.

Suppose, you have a million samples in your dataset, so if you use a typical Gradient Descent optimization technique, you will have to use all of the one million samples for completing one iteration while performing the Gradient Descent, and it has to be done for every iteration until the minima is reached. Hence, it becomes computationally very expensive to perform.

This problem is solved by Stochastic Gradient Descent. In SGD, it uses only a single sample, i.e., a batch size of one, to perform each iteration. The sample is randomly shuffled and selected for performing the iteration.

$$for\ i\ in\ range\ (m):$$
$$\theta_j = \theta_j - \alpha\,(\hat{y}^i - y^i)\,x_j^i$$

So, in SGD, we find out the gradient of the cost function of a single example at each iteration instead of the sum of the gradient of the cost function of all the examples.

In SGD, since only one sample from the dataset is chosen at random for each iteration, the path taken by the algorithm to reach the minima is usually noisier than your typical Gradient Descent algorithm. But that doesn't matter all that much because the path taken by the algorithm does not matter, as long as we reach the minima and with significantly shorter training time.

3.5 Training

A set of ML classifiers with various features has been used on our developed corpus, consisting two datasets 1. Twitter dataset (consisting of 163 thousand tweets) 2. Reddit dataset (37 thousand comments) where 50% documents used for training and 50% documents used for testing. The performance of the proposed system is compared with the human baseline and existing ML techniques.

Therefore a total of 210 thousand dataset entries used for training and testing purpose.

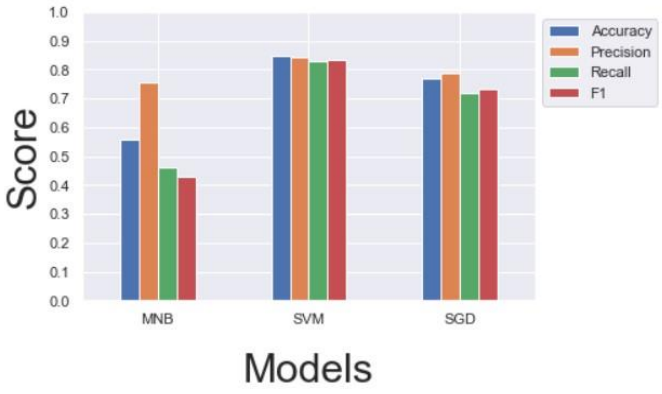

Chart -1: Visual Comparison of models for Twitter Dataset

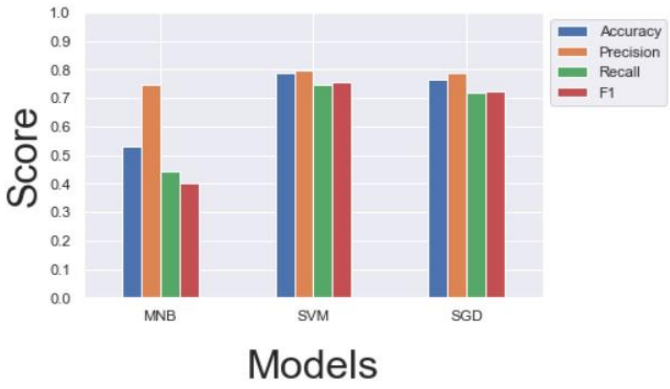

Chart -2: Visual Comparison of models for Reddit Dataset

4. CONCLUSION

Through this project, we learned about the concepts of NLP and how to extract features from text and train Machine Learning Models. If we directly extract features from raw text then the Machine Learning Models were giving poor performance. So we pre-processed the dataset before extracting features. We used Natural Language Processing techniques for that. After pre-processing of dataset we used Term Frequency-Inverse Document Frequency (TF-IDF) Vectorizer for feature extraction as Bag-of-Words method was giving importance to high frequency words whereas TF-IDF Vectorizer gave importance to the context of corpus. Once the features were extracted it was time to train models so we used different Machine Learning Models and used metric parameters such as Accuracy Score, Precision Score, Recall Score and F1 score. The models gave average results at first, but after shuffling the dataset the results improved

REFERENCES

[1] Murugesan, M. Sururthi, R. Pavitha Devi, S. Deepthi, V. Shri Lavanya, and Annie Princy. Automated Monitoring Suspicious Discussions on Online Formus Using Data Mining Statistical Corpus Based Approach. Imperial Journal of Interdisiplinary Research (IJIR) Vol-2, Issue-5, 2016

[2] Harika Upgaganlawar, Nilesh Sambhe. Surveillance of Suspicious Discussions on Online Forums Using Text Mining. International Journal of Advances in Electronics and Computer Science, Volume-4, Issue-4, April-2017

[3] Suhas Pandhe and Sahil Pawar. Algorithm to Monitor Suspicious on Social Networking Sites Using Data Mining Techniques. International Journal of Computer Applications. Volume 116 – No. 12, April 2015

[4] Javad Hosseinkhani, Mohammad Koochakazei, Solmaaz Keikhaee and Yahaya Hamedi Amin. Detecting Suspicion Information on Web Crime Using Crime Data Mining Techniques. International Journal of Advanced Computer Science and Information Technology (IJACSIT) Vol.-3, No. 1, 2014, Page 32-41

[5] G.Vinodhini, R.M Chandrasekran. Sentiment Analysis and Opinion Mining: A Survey. International Journal of Advanced Research in Computer Science and Software Engineering. Volume 2, Issue 6, June-2012

[6] Fabio Calefato, Filippo Lanubile, Nicole Novielli. EmoTxt: A Toolkit for Emotion Recognition from Text, University of Bari Aldo Moro

[7] M.F Portar. An Algorithm for Suffix Stripping Program. Vol. 14 Issue: 3, pp. 130-137