# Sentiment Analysis using Toxic Comment Classification

Nidhi S Sheth Prerana Prashant Kulkarni
Department of Computer Science & Engineering
PES University
Bangalore, Karnataka
Email: {nidhisheth007, preranakulk2003}@gmail.com

*Abstract*—**Sentiment Analysis is increasing gaining importance as social media platforms are now a source for identity theft and a medium to post hateful and toxic comments. In this paper, we aim to create user generated comments from Twitter into a dataset to train a deep learning sentiment analysis model that classifies the tweets to positive or negative tweets. In our approach, we used a bidirectional LSTM to obtain the desired outcome.**

***Keywords—deep learning, tweets, sentiment analysis, toxic comment classification, long-short memory, bi-LSTM***

## I. Introduction

In recent times, social media is gaining popularity as it provides a user-friendly platform for people to express their views, opinions and ideas to the world. It provides us with a user-friendly interface which makes it easier for people from various generations to use and participate in the same. Interactive apps and platforms like Facebook, Instagram, Twitter, LinkedIn are now being used for business collaborations and growth of emerging small businesses. Data generated from these sites are used by companies to monitor social media conversations and gain real-time insight into customers' opinions, experiences, likes and dislikes with brands using sentiment analysis tools [7]. Sentiment analysis has major real-world applications that include predicting the success of a new movie/song, determining the success of a new product put out in the market and many more. Social media has become very persuasive and can be easily misused by creating fake identity. Recently cyber-bullying and online harassment have become two of the most serious issues in many public online communities [10].

Toxic comments are disrespectful, abusive, or unreasonable online comments that usually make other users leave a discussion. The danger of online bullying and harassment affects the free flow of thoughts by restricting the dissenting opinions of people [9]. Social media companies are thus introducing policies to prevent online hatred with the help of sentiment analysis. It has proven to be a very useful and effective tool in preventing users from putting out hateful comments on their social media accounts.

Toxic comment classification was first done using Machine learning models. But soon deep learning models took over and proved to be more effective as it used word vectors with weights.[11]. Sentiment analysis has been categorized mainly into three levels- document level, sentence level and aspect level classification.

We made use of sentence level classification for our project.

Sentence level classification mainly focuses on detecting the sentiment polarity of a dataset on the sentence level by making use of the semantic information obtained from the content of the sentence.

## II. RELATED WORK

With the increased usage of social media apps and sites, people prefer to express their views and interact with each other through the same. This also led to increased number of cybercrimes and online hate. Twitter is one of the well-known social media websites where people express their opinions on a wide range of topics through tweets. We use sentiment analysis to classify these user generated comments into positive and negative tweets.

At first, sentiment analysis was done using machine learning algorithms like SVM, naive-Bayes, n-gram text classifiers. VADER is one such text classification technique we made use of for classifying our unlabeled dataset. VADER understands the polarity of a tweets having sarcasm, slang words, emoticons. It works excellent on social media text than other machine learning classifiers like text blob. [1]. But with further development Deep learning proved to be more effective than the traditional machine learning model when it comes to tweets as tweets are a lot different than normal comments as they include sarcastic statements that make determination of the tweet polarity difficult. Tweets are also written by making use of slang words that the machine learning techniques recognized with lesser effectiveness than the deep learning models like LSTM and BERT (State of the Art method). Moreover, a deep learning model does the same task with lesser computation cost and time. These models tokenize the tweets and convert them into smaller chunks and they are then fed to the model as vectors carrying weights. In this section we present the literature survey of deep learning models for toxic comment classification.

Any sentiment analysis model gives better results on balanced dataset that it gives with unbalanced dataset. Balanced dataset trains the dataset with equal number of positive and negative texts, thus improving its semantic knowledge. Mai Ibrahim used unique word, random mask and synonyms replacement text augmentation technique to duplicate the tweets from the label containing lesser tweets rather than using a balanced subset of the dataset and scraping the rest. This increased the accuracy from 0.8465 to 0.8825.

When compared to Support Vector Mechanism (SVM), Long Short-Term Memory (LSTM) gives better accuracy. LSTM remembers the important contents from the words it visited before and stores it in its long-term memory. This along with short term memory helps the model to learn the features with better accuracy than the SVM model that uses only short-term memory for semantic study. LSTM takes word embeddings (representation of a tweet as a vector) as input also outperforming BoW and n-gram that suffer from data sparsity [10]. An extension of LSTM is called bidirectional LSTM (BiLSTM) that makes use of two LSTM cells – forward LSTM and backward LSTM. This feature helps it to understand the semantic and meaning of a tweet in a much better way than LSTM as it captures the full context information.

This Bi-LSTM model can further be improved by adding additional layers. An attention layers makes the model detect the polarity and also capture important information about the aspect term in the sentence. This feature addition reduces model complexity due to vector splicing (transferring elements).[3]

A deep attention network t-DAN is used for detecting stance (what a person favors) on a target given. Here the attention layers check how close the target and tweets tokens are.[5]

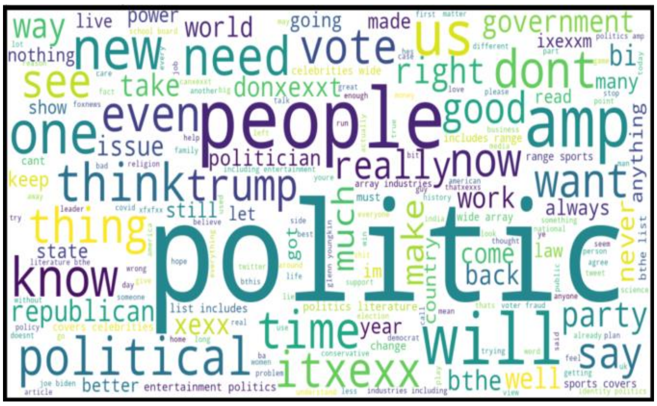
From the above literature survey, we chose to use the Bi-LSTM as our classification model. We fed clean tweets as the model input and were able to classify the tweets into positive and negative tweets. The detailed model framework is as shown below.

## III. IMPLEMENTATION

In this paper, we will be working on tweets. We aim to classify the tweets into positive and negative statements using Bidirectional LSTM.

### A. CREATING DATASET

For the creating of dataset, we used a python module Tweepy which allows easy access to the Twitter API. Scraping the tweets was done based on the hashtag “politics”. We managed to create a dataset that has 6209 tweets. The dataset contains the tweets, timestamp at the time of generation, the user profile name, and the follower’s count. In order to label the dataset, we need to process and free the tweets from hashtags, numeric values etc.



Wordcloud of politics dataset

### B. PRE-PROCESSING

Data preprocessing is essential before its actual use. Preprocessing converts raw data into clean data Extracted tweets will contain non-useful words like hashtags, mentions (@), words that do not belong to ‘utf-8’ and so on.

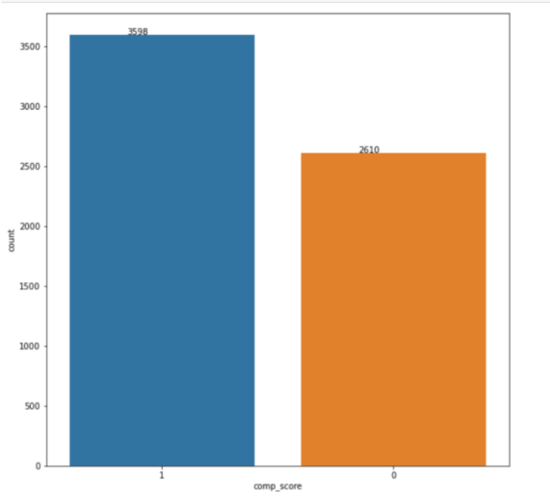
Thus, dataset is preprocessed in order to remove noisy data and other inconsistencies before use it in an algorithm. To obtain clean tweets, we do the following:

- Convert all the tweets into lower case to maintain uniformity.
- Remove the numerical values, tags/mentions, hashtags, any special characters, parenthesis and texts contained inside the parenthesis. This is done by replacing the same with a blank character using the python regular expression library.
- Removing stop words from the tweets. The most common stop words are ‘the’, ’am’, ’in’, ‘is’ etc. These words take up space in our dataset and are of no use in sentiment detection. Stop words from Natural Language Toolkit are removed from the dataset.
- Stemming and Lemmatization. Stemming removes the suffix of words and coverts them into root words. (E.g.: write, writer, writing gets converted to writ). Lemmatization converts the words into base or dictionary form called lemma.

### C. CLASSIFICATION OF TEXT

The dataset created using tweepy is unlabeled. classifying the tweets is important in order for the model to train and test on the same. There are various text classifiers available such as text blob, TF-IDF, VADER etc. We used VADER for text classification as it works very well on social media texts containing slangs, emoticons, conjunctions etc. and can be used on unlabeled data Here we used the nltk VADER lexicon model that is sensitive to polarity and strength of emotions. Tweets are classified into pos if compound value is greater than 1, using sentiment intensity analyzer.

Visualization:



Our politics dataset contains 42% negative tweets and 58% positive tweets which is pretty close to a balanced dataset.

### *D. MODEL ARCHITECTURE*

Sentiment analysis or toxic comment classification can be classified into 2 major categories: binary classification and multi-class classification. The model architecture is as explained below.

Our sentiment analysis model comprises of – an input layer, an embedding layer, a BiLSTM layer, dense layer, dropout layer and Rectified Linear Unit (reLU) layer at the output.

#### A. INPUT LAYER

The input layer is considered as a starting point of the network. Let w1, w2, w3, . . ., wn be the total number of unique words. The input layer carries data samples as a sequence of unique indices of same length.

#### B. EMBEDDING LAYER

This layer requires that the input data should be integer encoded, so that each word is represented by a unique integer. Using a Single-Layered BiLSTM Model unique word in the data set, is transformed into a vector. These vectors are stacked together to form a matrix, called an embedding matrix. In our paper, we used a ‘glove.6B.50d.txt’ as a pre-trained word embedding vector. It should have 3 parameters namely input_dim, output_dim and input_length.

#### C. Bi-LSTM LAYER

The most important layer is the bidirectional LSTM layer where the bidirectional long-term vectors between time series or sequence data are noted. These dependencies can be useful when you want the network to learn from the complete time series at each time step taking the required input from the input layer.
It sets additional states and parameters using one or more name-value pair arguments. You can specify multiple name-value pair arguments. Enclose each property name in quotes.

#### D. DENSE LAYER

Dense implements the operation: output = activation (dot (input, kernel) + bias) where activation is the element-wise activation function passed as the activation argument. These are all attributes of Dense.

#### E. DROPOUT LAYER

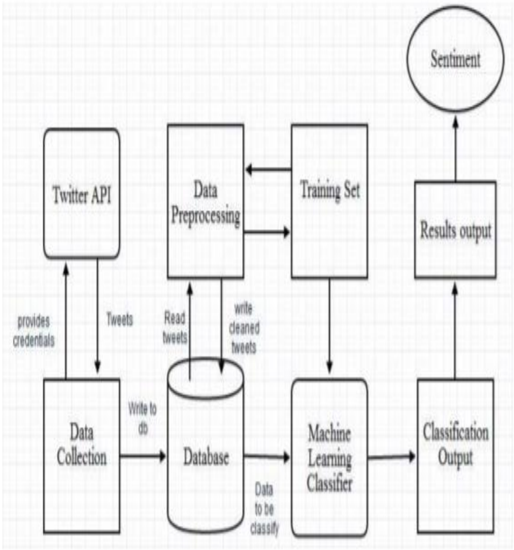
This layer helps prevent overfitting. Here certain tweets chosen randomly are ignored for each iteration by setting the input units to 0. This allows the layer to learn all the tweet semantic and improves generalised study of the tweet content and facilitating improvised training.

#### F. OUTPUT LAYER

reLU and Sigmoid are the most commonly used activation functions in the output layer. Since reLU is faster in calculation of its derivative, for the final classification we use a output layer with reLU activation function for the hidden layer with sparse categorical crossentropy as the loss function.

.

## IV DATA FLOW DIAGRAM



## IV. CONCLUSION

There are several metrics to calculate the model efficiency- precision, accuracy, recall, F1 score to name a few. Since we tested the model on an unbalanced dataset, we made use of the F1 metric. In statistical analysis, the F-score or F-measure is used as the measure of accuracy of the deep learning model. F score simply combines both precision and recall into a single value defining the effectiveness of the model on our particular dataset. F1 score can be calculated using

F1=2*(precision*recall)/(precision + recall)

Our single layered Bi-LSTM model trained and tested on the politics dataset containing around 6000 entries achieved an F1 score of 0.81 and an accuracy of 78%.

The future prospect of this study would be adding more layers into the Bi-LSTM model and introducing aspect level classification of text using attention layer. We would also like to use data augmentation technique to create a balanced dataset yielding better results.

## REFERENCES

[1] Mohamed Chiny , Marouane Chihab , Younes Chihab, Omar Bencharef "LSTM, VADER and TF-IDF based Hybrid Sentiment Analysis Model" VOLUME 2,2021, IJACSA

[2] ZABIT HAMEED AND BEGONYA GARCIA-ZAPIRAIN," Sentiment Classification Using a Single-Layered BiLSTM Model", VOLUME 8, 2020 pp:7992

[3] JUN XIE, BO CHEN, XINGLONG GU, FENGMEI LIANG AND XINYING XU "Self-Attention-Based BiLSTM Model for Short Text

Fine-Grained Sentiment Classification", VOLUME 7, 2019, PP:180558 Wei Yen Chong, Bhawani Selvaretnam, Lay-Ki Soon, "Natural Language Processing for Sentiment Analysis an Exploratory Analysis on Tweets", 2014 IEEE, pp:212

[4] Aditya Singh, Avinash kumar, Nishtha Dua, Vipul Kumar Mishra, Dilbag Singh, Apeksha Agrawal, "Predicting Elections Results using Social Media Activity A Case Study: USA Presidential Election 2020", 2021 IEEE, pp:314

[5] Yuanyu Yang, Bin Wu, Kai Zhao, Wenying Guo, "Tweet Stance Detection: A Two-stage DC-BILSTM Model Based on Semantic Attention", 2020 IEEE, pp:22

[6] YONGFENG DONG, YU FU, LIQIN WANG, YUNLIANG CHEN, YAO DONG, AND JIANXIN LI "A Sentiment Analysis Method of Capsule Network Based on BiLSTM", VOLUME 8, 2020, pp:37014

[7] Muhammad Husnain, Adnan Khalid, Numan Shafi, "A Novel Preprocessing Technique for Toxic Comment Classification", Artificial Intelligence (ICAI) ,2021 IEEE, pp:22

[8] [8] Rahul, Harsh Kajla, Jatin Hooda, Gajanand Saini, "Classification of Online Toxic Comments Using Machine Learning Algorithms", IEEE Xplore Part Number: CFP20K74-ART; ISBN: 978-1-7281-4876-2, pp:1119

[9] Mai Ibrahim, Marwan Torki and Nagwa El-Makky, "Imbalanced Toxic Comments Classification using Data Augmentation and Deep Learning", 2018 17th IEEE International Conference on Machine Learning and Applications, pp:875.

[10] Pumrapee Poomka, Nittaya Kerdprasop, and Kittisak Kerdprasop "Machine Learning Versus Deep Learning Performances on the Sentiment Analysis of Product Reviews.",2021 International Journal of Machine Learning and Computing, Vol. 11, No.2"

[11] Md Shofiqul IslamNgahzaifa Ab GhaniMd Manjur AhmedA "REVIEW ON RECENT ADVANCES IN DEEP LEARNING FOR SENTIMENT ANALYSIS: PERFORMANCES, CHALLENGES AND LIMITATIONS",2020 An international journal of advanced computer technology. Volume-IX, Issue-VII