ANALYSING TOXICITY OVER USER PROFILES USING DEEP LEARNING

A PROJECT REPORT

Submitted by

RAVIKUMAR S 2018115088

SARANYA V 2018115097

VETHAMATHURA S 2018115126

submitted to the Faculty of

INFORMATION AND COMMUNICATION ENGINEERING

in partial fulfillment for the award of the degree

of

BACHELOR OF TECHNOLOGY

in

INFORMATION TECHNOLOGY

DEPARTMENT OF INFORMATION SCIENCE AND TECHNOLOGY

COLLEGE OF ENGINEERING, GUINDY

ANNA UNIVERSITY

CHENNAI 600 025

JULY 2022

ANNA UNIVERSITY

CHENNAI - 600 025

BONA FIDE CERTIFICATE

Certified that this project report titled ”ANALYSING TOXICITY OVER USER PROFILES USING DEEP LEARNING ” is the bona fide work of ”RAVIKUMAR.S(2018115088)”, ”SARANYA.V(2018115097)”, ”VETHAMATHURA.S(2018115126)” who carried out project work under my supervision. Certified further that to the best of my knowledge and belief, the work reported herein does not form part of any other thesis or dissertation on the basis of which a degree or an award was conferred on an earlier occasion on this or any other candidate.

PLACE:	**Dr. L. SAIRAMESH**
DATE:	**TEACHING FELLOW**
	PROJECT GUIDE
	DEPARTMENT OF IST, CEG
	ANNA UNIVERSITY
	CHENNAI 600025

COUNTERSIGNED

Dr. S. SRIDHAR
HEAD OF THE DEPARTMENT
DEPARTMENT OF INFORMATION SCIENCE AND TECHNOLOGY
COLLEGE OF ENGINEERING, GUINDY
ANNA UNIVERSITY
CHENNAI 600025

ABSTRACT

The aim of this project is to develop a functional classifier for accurate and automatic sentiment classification of a web application. In the past decade, new forms of communication, such as microblogging have emerged and become ubiquitous. While there is no limit to the range of information conveyed by comments and texts, often these short messages are used to share opinions and sentiments that people have about what is going on in the world around them. This work performs 2 task for classifying a text to categorize the message is of positive, negative, or neutral using BERT, Naive Bayes, LSTM and NLP techniques. For messages, conveying both a positive and negative sentiment, whichever is the stronger sentiment should be chosen. Based on the sentiment analysis, toxic detection was carried out for analysing the toxicity over user profiles using deep learning process. The toxic detection consists of six categories such as toxic, severe toxic, obscene, threat, insult and identity hate. These are calculated in each tuple of a feature set using Logistic Regression, KNN, MultinomialNB, BernoulliNB, Linear SVC and Random Forest Classifier. Finally, the sentiment score and toxicity level of the comments is displayed in the web application for each and every comments of the post. Based on the threshold obtained, the user account was acknowledged over the nature of the post as toxic or not with the toxicity category.

ACKNOWLEDGEMENT

We express our profound sense of gratitude to our guide reassuring Dr. L. SAIRAMESH, Teaching Fellow, Department of Information Science and Technology, College of Engineering Guindy, Anna University, for his invaluable support and guidance for the successful completion of the project. His vision and spirit will always inspire and enlighten us.

We express our sincere thanks to Dr. S. SRIDHAR, Head, Department of Information Science and Technology, College of Engineering Guindy, Anna University, for the prompt and limitless help in providing the excellent computing facilities to do the project and to prepare the thesis.

We express our heartfelt gratitude to our review committee members, Dr. J. Indumathi, Professor, Dr. S. Swamynathan, Professor, Dr. P. Geetha, Assistant Professor, Dr. Selvi Ravindran, Assistant Professor, Ms. S. Kanimozhi, Teaching fellow, Ms. G. Mahalakshmi, Teaching fellow and Ms.B. Siva shankari, Teaching fellow, Department of Information Science and Technology, College of Engineering Guindy, Anna University, for their endless support during our project presentations. We express our heartiest thanks to all other teaching and non-teaching staffs those who have helped us to do the project and to prepare the thesis.

Finally, we acknowledge and thank our parents, who as ever remain our inspiration and strength, for their unyielding support and reassurance.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS AND ABBREVIATIONS

@	Ampersat, arobase, asperand, at, or at symbol.
API	Application Programming Interface
BERT	Bidirectional Encoder Representations from Transformers.
GloVe	Global Vectors for word representation.
KNN	K-Nearest Neighbor
LSTM	Long Short-Term Memory.
RNN	Recurrent Neural Network.

CHAPTER 1

INTRODUCTION

Micro-blogging websites have evolved to become a source of varied kind of information. This is due to nature of micro-blogs on which people post real time messages about their opinions on a variety of topics, discuss current issues, complain, and express positive sentiment for products they use in daily life. In fact, companies manufacturing such products have started to poll these micro-blogs to get a sense of general sentiment for their product. Many times these companies study user reactions and reply to users on micro-blogs. One challenge is to build technology to detect and summarize an overall sentiment. In this project, we look at one such web application for the desired output.

1.1 OVERVIEW

The primary purpose of social media site is to connect people and allow people to share their thoughts with a big audience. It uniquely provides its users the opportunity to discover what's happening in the world. That functional purpose translates into web app users operating in a different mindset when they're on the platform compared to other online environments. In particular, the act of discovery, the predominant way in which the audience uses the platform, puts them in a receptive mindset, meaning that information is stored in their long-term memory more effectively. This is particularly important for advertisers, as academic literature shows that long-term memory encoding relates closely to purchase intent. When it comes to brands, social media users are demanding consumers, who want quality and responsiveness, and they strongly reward brands that make the effort. This proves, social media audience is curious, leaned in and ultimately receptive. Combined with their influence, proclivity to share what they know, and loyalty to brands offering quality products in a way that align with their values, they are the perfect catalyst for any marketers campaign messaging.

1.2 NEED FOR SENTIMENT ANALYSIS

Nowadays, when the micro-blogging platforms are commonly used, the task of sentiment analysis becomes even more interesting. Although there are several known tasks related to sentiment analysis, in this project we will focus on the common binary problem of identifying the positive / negative sentiment that is expressed by a given text toward a specific topic. In other words, the texts that we deal with in this project, must express either positive or negative sentiment, and they cannot even be neutral about the topic. There are other tasks that allow a text to be neutral about a specific topic, or even totally objective, i.e. expressing no interest in the topic at all. For this task, we explore models for classifying “comments” into positive, negative and neutral sentiment using two types of models: BERT based model and LSTM based model.

1.3 DATASET

Web scraping is the process of extracting data from a website and exporting it in a manner that is more helpful to the user. Here, we created a web application called Toxic Detection with Comments to collect data for the project.

And to begin our first step, there are many things to consider when choosing how to preprocess your text data, but before you do that you will need to familiarize yourself with your data. This dataset is provided in a .csv file and it is loaded into a dataframe and proceeded with the further process. Before data preprocessing, there are three main tasks in text preprocessing to do such as case-folding, filtering, tokenization, normalization and stopwords removal.

1.4 NEED FOR TOXIC DETECTION

The hyperconnectivity offered by the internet in a world where there are people with various culture, educational background and opinions represents a challenge for healthy social interactions. Toxic and hostile content has become a major problem in most social networks or forums and the task of controlling it has overcome non-automated resources. The key for preventing the spreading of this toxic content is improving automatic detection of these social media patterns. In this work, we use machine learning classifiers for the detection of comments namely LSTM-RNN.

1.5 CHALLENGES IN COMMENTS ANALYSIS

When it comes to analysis challenges, there are quite a few things that we struggle with in order to obtain sentiment analysis accuracy. Sentiment or emotion analysis can be difficult in natural language processing simply because machines have to be trained to analyze and understand emotions as a human brain does.

1. **Polarity:** Words such as “love” and “hate” are high on positive (+1) and negative (-1) scores in polarity. These are easy to understand. But there are in-between conjugations of words such as “not so bad” that can mean “average” and lie in mid-polarity (-75). Sometimes phrases like these get left out, which dilutes the sentiment score.
2. **Emojis:** The problem with social media content is that, the texts are inundated with emojis. NLP tasks are trained to be language specific.

While they can extract text from even images, emojis are a language in itself. Most emotion analysis solutions treat emojis like special characters that are removed from the data during the process of text mining.

3. **Tone:** Tone can be difficult to interpret verbally, and even more difficult to figure out in the written word. Things get even more complicated when one tries to analyze a massive volume of data that can contain both subjective and objective responses. Brands can face difficulties in finding subjective sentiments and properly analyzing them for their intended tone.

4. **Idioms:** Machine learning programs don't necessarily understand a figure of speech. For example, an idiom like "not my cup of tea" will boggle the algorithm because it understands things in the literal sense. Hence, when an idiom is used in a comment or a review, the sentence can be misconstrued by the algorithm or even ignored. To overcome this problem a sentiment analysis platform needs to be trained in understanding idioms. When it comes to multiple language, this problem becomes manifold.

5. **Negations:** Negations, given by words such as not, never, cannot, were not, etc. can confuse the ML model. For example, a machine algorithm needs to understand that a phrase that says, "I can't not go to my class reunion", means that the person intends to go to the class reunion.

6. **Comparative sentences:** Comparative sentences can be tricky as they may not always give an opinion. Much of it has to be deduced. For example, when somebody writes, "the Galaxy S20 is larger than the Apple iphone12", the sentence does not mention any negative or positive emotion but rather states a relative ordering in terms of the

size of the two phones.

7. **Multilingual sentiment analysis:** It constitutes all the problems listed above get compounded when a mix of languages are thrown in. Each language needs a unique part-of-speech tagger, lemmatizer, and grammatical constructs to understand negations.

1.6 PROBLEM STATEMENT

In this project, we are implementing the sentiment and toxicity analysis model that helps to overcome the challenges of identifying the sentiments and toxicity of the comments.

When a user adds the post to the application, the other users can view, like and post the comments for the posts and it will be analysed using machine learning and deep learning classifiers techniques and acknowledge the user whether the particular post is toxic or not.

1.7 OBSERVATION AND MOTIVATION

The reason is that the amount of relevant data is much larger for social media sites, as compared to traditional blogging sites. Moreover the response on social media is more prompt and also more general. Sentiment analysis of public is highly critical in macro-scale socio-economic phenomena like predicting the stock market rate of a particular firm. This could be done by analysing overall public sentiment towards that firm with respect to time and using economics tools for finding the correlation between public sentiment and the firm's stock market value.

1.8 ORGANIZATION OF THE PROJECT

The rest of the report is organized as follows:

- Chapter 2 discusses about the Literature survey which summarizes the related works that had been published related to the current system and describes how this system is enhanced to overcome the limitations of the existing system
- Chapter 3 discusses about the System architecture which describes the overall work flow, with detailed explanation of the modules in the architecture diagram.
- Chapter 4 discusses the details of implemention with the algorithms which discusses the procedure and workflow of implementation.
- Chapter 5 emphasizes the results of this system with screenshots of the output.
- Chapter 6 discusses the conclusion and future enhancements of this system

CHAPTER 2

LITERATURE SURVEY

2.1 EXISTING WORKS

The modules of the system are analyzed with the help of articles and journals and these references are discussed below

Majed Alrubaian et al., [1] proposed a system for sentiment analysis on twitter. The proposed system extracted the data from the twitter using the twitter streaming API. The twitter streaming API also includes search option for searching the particular tweets based on hashtags, event specific (related to particular topic), etc.

Aditi Gupta and Ponnurangam Kumaraguru [2] proposed a system that involves in extracting tweets baesd on query terms by using trends API from twitter, which returns top 10 trending topics on twitter.

Rim El Ballouli et al., [3] proposed a system involves in collection and processing of tweet data by collecting tweets form the reputable sources and also collecting both tweets and news articles on trendy topics.

Takashi Kawabe et al., [4] proposed a method that recognize the positive or negative opinion in the tweet. In this phase, takamura's semantic orientation dictionary is used. This dictionary uses a single word surface and part-of-speech analysis and builds so called meaning clusters and derives a positive, negative, or neutral semantic polarity value from it. The values range from -1 to 1.

C. J. Hutto et al., [5] proposed a methodology on monitoring user growth for posting the topic specific tweets in twitter over a period of time. Topic specific tweets can be measured by predictor variables such as topic focus, informational content index, hashtags etc.

Wei Wei et al., [6] focuses on identifying people with the relevant expertise or experiences on a given topic and also focuses on topic specific expert finding in twitter.

Cao and Rei., [7] proposed a novel approach named char2vec based on the representation of characters instead of words.

Ali et al., [8] proposed a system that retrieved transport content from social networks, representing the documents with word embedding techniques and achieving an effective approach to sentiment classification with 93 percent accuracy.

Bollen et al., [9] measures positive and negative moods and Google - Profile of Mood States (GPOMS) that measures mood in terms of 6 dimensions (Calm, Alert, Sure, Vital, Kind, and Happy) to predict the changes in DJIA closing values of Twitter text using mood tracking tools. They achieved an accuracy of 86.7 percent in predicting the daily up and down changes in the closing values of the DJIA stock market index.

Jacob Devlin, Ming-Wei Chang, Kenton Lee and Kristina Toutanova., [10] have demonstrated unsupervised pre-training is an integral part of many language understanding systems.And further generalizing these findings to deep bidirectional architectures to successfully tackle a broad set of NLP tasks.

Xin Li , Lidong Bing , Wenxuan Zhang and Wai Lam., [11] explored the effectiveness of BERT embedding component on the task of Endto-End Aspect-Based Sentiment Analysis. The experimental results demonstrate the superiority of BERT-based models on capturing aspect-based sentiment and their robustness to overfitting.

Apoorv Agarwal, Boyi Xie ,Ilia Vovsha, Owen Rambow and Rebecca Passonneau., [12] presented results for sentiment analysis on Twitter. Here, they investigated two kinds of models: tree kernel and feature based models and demonstrated both these models outperform the unigram baseline. For feature-based approach, feature analysis is done which reveals the most important features that combine the prior polarity of words and their parts-of-speech tags. And it was tentatively concluded that, sentiment analysis for Twitter data is not that different from sentiment analysis for other genres.

Bouazizi M and Ohtsuki T., [13] have proposed a new approach for sentiment analysis, where a set of tweets is to be classified into 7 different classes and the accuracy obtained for multi-class sentiment analysis in the data set used was 60.2 percent. However, a more optimized training set would present better performances.

Eider Diaz, Miguel Cortes, and Carlos Hinojosa.,[14] presented 4 classifiers trained for hate-speech and offensive language detection using Twitter data. Those models suffered from class imbalance as the recall for the hateful and normal language categories in all models are very low. However, the results still reflect the overall behaviour of this subset of Twitter users as the results are consistent with what users have reported and expressed about this specific community.

Davidov, Tsur and Rappoport., [15] used hashtags to create training data and limited the experiment to sentiment/non-sentiment classification rather than 3-way polarity classification.

2.2 LIMITATIONS OF EXISTING WORK

From the study of literatives, the following points are considered as the limitations of existing work.

- Computer programs have problems recognizing things like sarcasm and idioms, negations, jokes, and exaggerations - the sorts of things a person would have little trouble identifying. And failing to recognize these can skew the results.
- The desired output is not obtained in case of special characters and emoji's attached with the text messages or images.
- The problem of word ambiguity is the impossibility to define polarity in advance because the polarity for some words is strongly dependent on the sentence context.
- Inaccurate representation of overall sentiments and earlier models does not analyse any wide range of influence ties between the users.
- Existing methodologies did not considered measures that are easily generalizable such as favourites, likes etc.

2.3 PROPOSED WORK

This section describe the proposed model for the tasks of emotion and sentiment analysis. The model is built by fine-tuning BERT on specific data sets of web application developed for sentiment analysis. The raw data is collected from the web application, generally result in a very noisy and obscure Dataset, due to people's random and creative use of social media. Comments have certain special features, i.e., emojis, hashtags and user mentions, coupled with typical web constructs, such as email addresses and URLs, and other noisy sources. Since comments usually contain words that are irrelevant for text classification, a text pre-processing phase is needed in order to remove re-comments, url's and mentions. After the pre-processing phase, data can be used as input to train task-specific BERT-based models (i.e, 75 percent data for training and 25 percent data for testing). Based on the attitudes and emotions of the comments, the toxic level of the comment is analysed using LSTM-RNN models and the output is stored in the database. Then the threshold value is fixed for the process. Based on that, the user is acknowledged over their posts as toxic or not.

2.4 OBJECTIVES

- To extract the comments from web app, for the post of user profiles.
- To preprocess the comments and remove the stop words, urls, and unwanted comments using NLP(natural language processing).
- To classify the positive and negative comments using BERT model.
- To analyze the account details of the user such as followers, mentions, impressions, followings etc for finding the most influence user.
- To identify the toxicity of the users comments for the posts using LSTM-RNN model and generate word cloud for sentiment analysis and toxic detection with different color-code.

2.5 COMPARATIVE STUDY

Comparative study of the literature survey is shown in Table 2.1

Table 2.1: SUMMARY OF LITERATURE SURVEY.

PUBLISHING/YEAR	MECHANISM	PROS	CONS	DATASET
2020 International Conference of Modern Applications on Information and Communication Technology	Extracts aspects and expressions using Sentiment Pattern Analyzer based on MSB Model	It does not need to make any model assumption as like in K-means or Gaussian mixture. It can also model the complex clusters which have nonconvex shape	Mean-shift algorithm does not work well in case of high dimension, where number of clusters changes abruptly.	Twitter API, You Tube, Movie Reviews
2015 International Conference on Computational intelligence and communication networks	Rapidminer- extracts information using keywords and Sentiword - used for assigning polarities	Multiple deployment options based on your preference and strong visualization with accurate preprocessing.	It takes too much memory and so slows down your system and has less forums for support and it is tough for new users.	Twitter API
2019 Sixteenth International Conference on Wireless and Optical communication networks	Opinion summarization using SVM and Naive Bayes	SVM works relatively well when there is a clear margin of separation between classes and it is more effective in high dimensional spaces.	SVM does not perform very well when the data set has more noise i.e. target classes are overlapping and is not suitable for large data.	General Data.

PUBLISHING/YEAR	MECHANISM	PROS	CONS	DATASET
2015 IEEE 14th International Conference on Cognitive Informatics and Cognitive Computing	Feed Forward Neural Network-reducing input sequence by removing @ and so on	Easy to conceptualize and lots of libraries implementation available	Multi-layer neural networks are usually hard to train and require tuning lots of parameters	Twitter API
2019 Fourth International Conference on Informatics and Computing	NLTK for tokenization and uses Naive Bayes for classification	Works with high dimensional data and with high scalability	Assumption of independent predictor features in real life makes the accuracy lesser than other complicated algorithms	Twitter API and tweets from michigan's sentiment analysis

CHAPTER 3

SYSTEM DESIGN

3.1 SYSTEM ARCHITECTURE

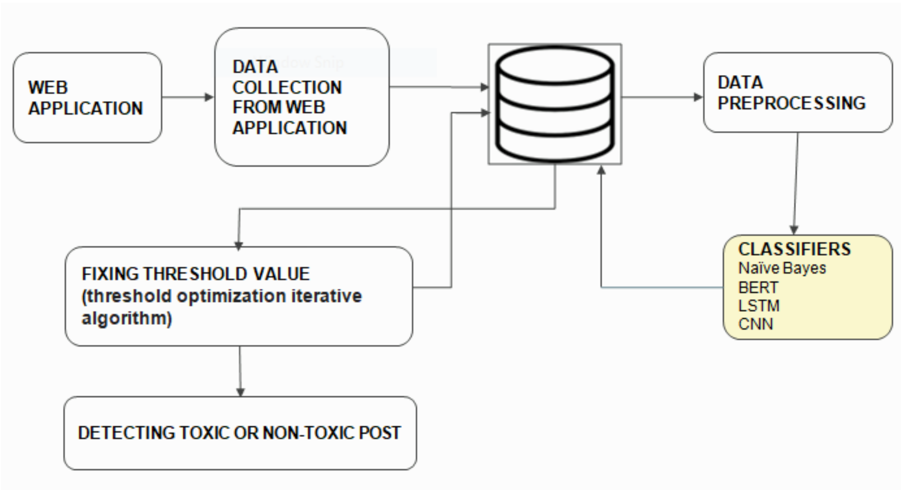


Figure 3.1: Architecture Diagram

Figure 3.1 shows the architecture for toxicity analysis of comments posted in web application.

3.2 MODULES

The following list is the modules of toxicity analysis system:

1. Data collection from web application
2. Data preprocessing
3. Machine learning model
4. Deep learning model

5. Perspective of toxic comments
6. Word cloud generation
7. Threshold value fixation

3.2.1 Data Collection from Web Application

Web scraping refers to the extraction of data from a website and that information is exported into a format which is more useful for the user. A web application was designed namely Toxic Detection with comments to gather data for further actions of our project. Basic features of the web application are as follows:

1. The users are asked to sign in with their google accounts to access the web application.
2. The users can upload the posts based on their interest and can even add comments and drop likes for the post of other users.
3. The likes, comments, post time, comment time and user accounts have been stored in the database and retrieved whenever it is required.

3.2.2 Data Preprocessing

Data preprocessing is the process of transforming raw data into an useful format. Some text formatting techniques which will aid us in data preprocessing:

- Tokenization: Tokenization is the process of breaking a stream of text up into words, symbols and other meaningful elements called “tokens”. Tokens can be separated by whitespace characters and/or punctuation characters. It is done so that we can look at tokens as individual components that make up a sentence.

- Url's and user references (identified by tokens "http" and "@") are removed if we are interested for analyzing the text of the web app.

- Punctuation marks and digits/numerals may be removed. For example, we wish to compare the comment text to a list of English words.

- Stemming: It is the text normalizing process of reducing a derived word to its root or stem. For example, a stemmer would reduce the phrases "stemmer", "stemmed", "stemming" to the root word "stem". Advantage of stemming is that it makes comparison between words simpler, as we do not need to deal with complex grammatical transformations of the word. In our case we employed the algorithm of "porter stemming" on both the textual comments and the dictionary, whenever there was a need of comparison.

- Stop-words removal: Stop words are class of some extremely common words which hold no additional information when used in a text and are thus claimed to be useless.

- Parts-of-Speech Tagging: POS-Tagging is the process of assigning a tag to each word in the sentence as to which grammatical part of speech that word belongs to, i.e. noun, verb, adjective, adverb, coordinating conjunction etc.

- Lemmatization: Lemmatization is the process of grouping together the different inflected forms of a word so they can be analyzed as a single item. Lemmatization is similar to stemming but it brings context to the words. So, it links words with similar meaning to one word. For example, runs, running, ran are all forms of the word run, therefore run is the lemma of all these words.

3.2.3 Machine Learning Model

3.2.3.1 BERT Model

BERT, or Bidirectional Encoder Representations from Transformers makes use of a transformer which is essentially a mechanism to build relationship between the words in the dataset. In its simplest form, a BERT consists of two processing models- an encoder and a decoder. The encoder reads the input text and the decoder produces the predictions. But, because the main goal of BERT is to create pre-trained model, the encoder takes priority over decoder. A sentence is first split into individual words and this is embedded into vectors. After converting the data in the required format, the next step is to train the tokenizer on input data. The transformer processes these vectors and produces outputs, which are also vectors in which each vector corresponds to an input token with the same index. For the fine-tuning section, the data must be in a different format from what we used in the pre-training part.

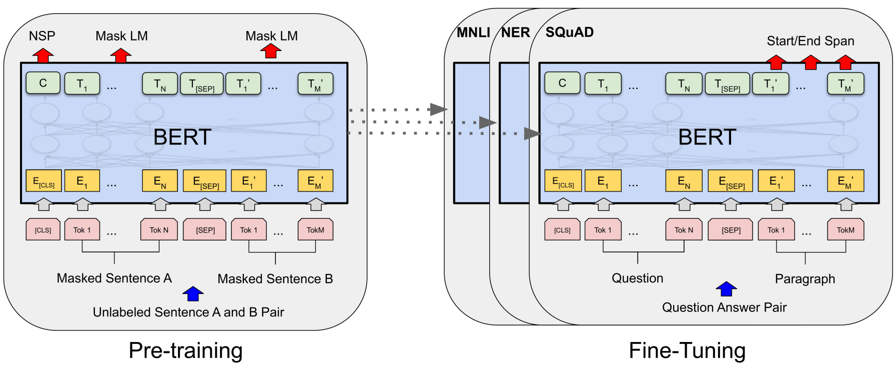


Figure 3.2: BERT Model

Figure 3.2 shows the BERT model sentiment analysis.

3.2.3.2 Naive Bayesian Classifiers

- All features are independent from each other.
- Every feature contributes equally to the output.

In our case, these two assumptions can be interpreted as:

- Each word is independent from the other words, no relation between any two words of a given sentence.
- Each word contributes equally for all sentences, to the decision of our model, regardless of its relative position in the sentence.

3.2.4 LSTM-RNN Model

Long Short-Term Memory (LSTM) networks are capable of learning order dependence in sequence prediction problems. Predicted sentiment using LSTM-RNN, which gives high accuracy, can be used to educate people about the negativities in tweets. This model will be able to find out the actual meaning in input string and will give the most accurate output class.

3.2.5 Perspective of Toxic Comments

- **Toxicity** score represents the degree to which the comment is rude, disrespectful, or unreasonable.
- **Severe Toxicity** score represents how hateful, aggressive, and rude the comment is.

- **Profanity** score indicates if swear words or other profane language is used.
- **Identity Attack** score indicates if a post contains hateful language targeting someone because of their identity.
- **Insult** score helps to identify insulting or inflammatory posts.
- **Threat** score represents the degree to which a post displays an intention to inflict pain or violence against an individual or group.
- **Sexually Explicit** score indicates if a post contains references to sexual acts, body parts, or other lewd content.
- **Flirtation** score indicates if a post contains language commonly used in pickup lines, compliments regarding appearance, or subtle sexual innuendo.

3.2.6 Word Cloud Generation

A word cloud (known as a tag cloud) is a visual representation of words. Cloud creators are used to highlight popular words and phrases based on frequency and relevance. They provide you with quick and simple visual insights that can lead to more in-depth analyses. In this project, Wordcloud is used for generating a list of words based on sentiments and they are color-coded in different colors. Each word's size of a word cloud represents how frequently is the word has been mentioned on Web application. After extracting the data from web application, sentiment for pharses is done, if the desired results are not produced then sentiment analysis for each words is done and finally based on the polarity they are color-coded.

3.2.7 Threshold Value Fixation

The threshold value is the choice to turn a projected probability or scores into a class label. For the normalized cases, the projected probability is in the range of 0 to 1, and sometimes it is set to 0.5 by default. In this project, based on the threshold value obtained using threshold optimization iterative algorithm the user posts are analysed and if the frequency is highly negative, then the users are acknowledged regarding the post.

3.3 EVALUATION METRICS

Classification tasks for evaluating the polarities of the comments

$$\mathrm{F}_c = 2(P_c P_r)/P_c + P_r$$

Where P indicates polarity score of the comment
c indicates the class(0 or 1 or -1)
R indicates recall value of the class.
F indicates the score of each task

Average of each task is defined by the polarities of the comment

$$\mathrm{F} = \mathrm{F}_1(pos) + F_2(neg)/2$$

BERT confusion matrix for the sentiment analysis task is:

category	RECALL Value	PRECISION Value
Positive	0.91	0.96
Negative	0.96	0.84
Neutral	0.90	0.95

CHAPTER 4

IMPLEMENTATION

4.1 CONTENTS OF WEB APPLICATION

Web application is build using React JS, nextjs, tailwind css, nextauth, recoil and firebase v9. It is designed for social media users to have a better experience and information about their comments that is being posted for the posts of other users with the levels of toxicity and sentiment detection.

4.2 DATA EXTRACTION AND PREPROCESSING

The process of data extraction from web application is done and the raw data is stored in the database for further processes. Before that there is a need for preprocessing (i.e, the data undergoes tokenization, stemming and lemmatization). Word2Vec algorithm and punkt tokenizer is used for this purpose.

Punkt tokenizer divides a text into a list of sentences by using an unsupervised algorithm to build a model for abbreviation words, collocations, and words that start sentences. It must be trained on a large collection of plaintext in the target language before it can be used. The NLTK data package includes a pre-trained Punkt tokenizer for English.

Algorithm 4.1 Word2Vec Algorithm

Input : Comments with stopwords and special characters.
Output : Word tokens.

1. Initially, we will assign a vector of random numbers to each word in the corpus.

2. Then, we will iterate through each word of the document and grab the vectors of the nearest n-words on either side of our target word, and concatenate all these vectors, and then forward propagate these concatenated vectors through a linear layer + softmax function, and try to predict what our target word was.

3. , Herewe will compute the error between our estimate and the actual target word and then back propagated the error and then modifies not only the weights of the linear layer but also the vectors or embeddings of our neighbor's words.

4. Finally, we will extract the weights from the hidden layer and by using these weights encode the meaning of words in the vocabulary

Algorithm 4.2 Punkt tokenizer Algorithm

Input : character sequence.
Output : word tokens.

1. Split the words in the corpus into characters after appending ¡/w¿

2. Initialize the vocabulary with unique characters in the corpus.

3. Compute the frequency of a pair of characters or character sequences in corpus.

4. Merge the most frequent pair in corpus.

5. Save the best pair to the vocabulary.

6. Repeat steps 3 to 5 for a certain number of iterations.

Stemming is the process of reducing the word to its word stem that affixes to suffixes and prefixes or to roots of words known as a lemma. In this project we use snowball stemming for sentiment analysis and lancaster stemming for toxicity detection. The stepwise procedures are common for both the stemming algorithms the only variant is the snowball stemming can map non-English words too and sufficient for small words whereas the lancaster stemming is more aggressive and dynamic and slightly confusing for small words.

Algorithm 4.3 Stemming Algorithm

Input : Tokenized words.
Output : Reduced root words.

1. Convert the plural form of a word to its singular form.

2. Convert the past tense of a word to its present tense and remove the suffix 'ing'.

3. Compare the words with the vocabulary.

4. The output of the word will be the lemma.

Here, we used WordNet Lemmatizer to get the base form of all the words having same meaning.

Algorithm 4.4 Lemmatization Algorithm

Input : Tokenized words.
Output : Reduced root words.

1. Remove inflectional endings.

2. Finding the lemma of a word depending on its meaning and context.

3. Convert all words having the same meaning but different representation to their base form.

4.3 FEATURE EXTRACTION

The process of feature extraction involves in extracting the features of the comments such as stance, count, wordcount, charcount, average word, uppercase, lowercase, wordtokens and hashtags. The word tokens are obtained as output from the pre-processing module. The obtained feature set undergoes further processing in the upcoming stages.

Algorithm 4.5 TD-IDF Vectorizer to calculate test score for toxicity

---

Input : Comments with stopwords and special characters.
Output : Word tokens with frequency matrix.

1. Finding the frequency on each of the document [N(t, d)].

2. Finding the term frequency for a term t in document d is represented as:
$tf(t, d) = N(t, d)$

3. Finding the total number of term in the document (nD).

4. Normalized term frequency ntf(t, d) for a term t in document d is given as:
$ntf(t, d) = N(t, d)/mod(nD)$.

5. Final output will be the vector space representation of the document.

4.4 SENTIMENT ANALYSIS

For sentiment analysis we have used BERT algorithm with pipeline to classsify the comments as postive and negative.

Algorithm 4.6 Bert Model Algorithm

Input : Feature set with the above mentioned attributes.
Output : Sentiment score will be acquired.

1. BERT was pretrained on two tasks: language modelling (15 pecent of tokens were masked and BERT was trained to predict them from context) and next sentence prediction .

2. BERT was trained to predict if a chosen next sentence was probable or not given the first sentence.

3. As a result of the training process, BERT learns contextual embeddings for words.

4. After pretraining, which is computationally expensive, BERT can be finetuned with less resources on smaller datasets to optimize its performance on specific tasks.

4.5 TOXICITY DETECTION

Logistic regression, KNN, MultinomialNB, LinearSVC, BernoulliNB, and RandomForestClassifier are used for classification of positive and negative comments in a feature set and to detect the toxicity.

In Logistic Regression algorithm, a tweet or some text is given. we can represent it as a vector of dimension V, where V corresponds to vocabulary size. The idea is to divide the training set into positive and negative comments. Count all the words and make a python dictionary of their frequencies in positive and negative comments. For every comments make a vector of bias unit, sum of all the positive frequencies(words from positive comments) of all the words and also their negative frequencies.

Algorithm 4.7 Logistic Regression Algorithm

Input : Feature set with the above mentioned attributes.
Output : Toxicity score will be represented as binary matrix.

1. Convert comments(text) into frequency matrix(numeric) using Vectorizer.

2. X is the Frequency matrix and Y is the output (0 or 1).

3. Structuring the training data using logistic regression with sigmoid function.

4. Test accuracy of the result(Creation of Confusion matrix)

5. Visualizing the test set result.

K Nearest Neighbor algorithm for text classification with Python. KNN algorithm is used to classify by finding the K nearest matches in training data and then using the label of closest matches to predict. Traditionally, distance such as euclidean is used to find the closest match.

Algorithm 4.8 K-Nearest Neighbour Algorithm

Input : Feature set with the above mentioned attributes.
Output : visualization of data as class membership.

1. Select the number K of the neighbors

2. Calculate the Euclidean distance of K number of neighbors.

3. Take the K nearest neighbors as per the calculated Euclidean distance.

4. Among these k neighbors, count the number of the data points in each category.

5. Assign the new data points to that category for which the number of the neighbor is maximum.

6. If regression, return the mean of the K labels.

7. If classification, return the mode of the K labels.

Random Forest Classifier Algorithm: A single decision tree can learn quite complex functions. However , in many ways it can be prone to overfitting. To overcome this, we could create many decision trees and then ask each tree to predict the class value. We could take majority vote and use that answer as our overall prediction. Random forest work on this principle.

Algorithm 4.9 Random Forest Classifier Algorithm

Input : Feature set with the above mentioned attributes.
Output : Estimated probability to improve predictive accuracy of the dataset.

1. Select random K data points from the training set.

2. Build the decision trees associated with the selected data points (Subsets).

3. Choose the number N for decision trees that you want to build.

4. Repeat Step 1 2.

5. For new data points, find the predictions of each decision tree, and assign the new data points to the category that wins the majority votes.

Multinomial Naive Bayes is used when we have discrete data (e.g. comments ratings ranging 1 and 5 as each rating will have certain frequency to represent). In text classification we have the count of each word to predict the class or label. If the words can be represented in terms of their occurrences (frequency count) then use multinomial event model. If we just care about the presence or absence of a word in the document, then use Bernoulli event model.

Bernoulli Naive Bayes assumes that all our features are binary such that they take only two values. Means 0s can represent “word does not occur in the document” and 1s as “word occurs in the document”.

Algorithm 4.10 MultinomialNB Algorithm

TRAINMULTINOMIALNB(C,D)

1. $V \leftarrow EXTRACTVOCABULARY(D)$

2. $N \leftarrow COUNTDOCS(D)$

3. for each$c \in C$

4. do $N_C \leftarrow COUNTDOCSINCLASS(DC)$

5. $prior[c] \leftarrow N_c/N$

6. $text_c \leftarrow CONCATENATETEXTOFALLDOCSINCLASS(D,c)$

7. for each$t \in V$

8. do $T_{ct} \leftarrow COUNTTOKENSOFTERM(text_c,t)$

9. for each $t \in V$

10. do $condprob[t][c] \leftarrow \frac{T_{ct}+1}{\sum_{t'}(T_{ct'+1})}$

11. return V, prior, condprob

APPLYMULTINOMIALNB(C,V,prior,condprob,d)

1. $W \leftarrow EXTRACTTOKENSFROMDOC(V,d)$

2. for each $c \in C$

3. do $score[c] \leftarrow logprior[c]$

4. for each $t \in W$

5. do score[c] += log condprob[t][c]

6. return $argmax_{c \in C} score[c]$

Algorithm 4.11 BernoulliNB Algorithm

TRAINBERNOULLILNB(C,D)

1. $V \leftarrow EXTRACTVOCABULARY(D)$

2. $N \leftarrow COUNTDOCS(D)$

3. for each$c \in C$

4. do $N_C \leftarrow COUNTDOCSINCLASS(DC)$

5. $prior[c] \leftarrow N_c/N$

6. for each$t \in V$

7. do $N_{ct} \leftarrow COUNTDOCSINCLASSCONTAININGTERM(D,c,t)$

8. $condprob[t][c] \leftarrow (N_{ct}+1)/(N_c+2)$

9. return V, prior, condprob

APPLYBERNOULLINB(C,V,prior,condprob,d)

1. $V_d \leftarrow EXTRACTTERMSFROMDOC(V,d)$

2. for each $c \in C$

3. do $score[c] \leftarrow logprior[c]$

4. for each $t \in V$

5. do if$t \in V_d$

6. then score[c] += log condprob[t][c]

7. else score[c] += log(1-condprob[t][c])

8. return $argmax_{c \in C} score[c]$

Threshold optimization iterative algorithm for binary classification refers to the practice of fixing a threshold t and then labeling predictions above it as positive and below it as negative. In this binary classification problem, normalized predicted probabilities and a threshold of 0.4, then values less than the threshold of 0.4 are assigned to negative, values greater than 0.4 and lesser than 0.6 are assigned to neutral and values greater than 0.6 are assigned to positive.

Prediction$\leq$0.4 = negative

0.4 $<$ Prediction $>$ 0.6 = neutral

Prediction $\geq$ 0.6 = positive

Algorithm 4.12 Threshold optimization iterative Algorithm

Input : Feature set with the above mentioned attributes.
Output : Threshold value.

1. Collection of data.

2. Selection of the information feature.
Entropy(y)=sum(Pi$\times log2(Pi)$)
$gain(y,A) = entropy(y)\sum_{C\in vals(A)} \frac{yc}{y} entropy(yc)$

3. Calculating the standard deviation of the information for each feature. The value obtained during the calculation of all features' standard deviation is known as the threshold. All features possessing a value greater than or equal to that of the threshold value should be selected. Those having a lesser value than that of the threshold should be discarded.

$$S=\sqrt{\frac{n\sum_{i=1}^{n} x_i^2 - (\sum_{i=1}^{n} x_1)^2}{n(n-1)}}$$

4. Performing the selection process, by removing features having an information value below that of the threshold.

5. The results of the feature selection were obtained.

CHAPTER 5

RESULTS AND PERFORMANCE ANALYSIS

To evaluate the toxicity analysing system by assessing the sentiment classifier and the toxic detection.

5.1 DATA AND TOXICITY ANALYSIS

Data for assessing sentiment analysis and toxic detection are extracted from the web application (Toxicity detection with comments). The training dataset consists of 10,662 comments of which 5331 comments are labeled as positive (1) and the remaining 5331 comments are labeled as negative (0). The test dataset consists of 2202 tuples of comments out of which 1242 comments are positive (1) comments and 960 comments are negative (0) comments. After testing, the sentiment of comments are calculated and combined with the feature set. A feature set comprises of 4 attributes such as Comments, Likes, Time, Source. After testing, the toxicity of comments are calculated and combined with feature set. Here , the feature set consists of 8 attributes such as id, comment text, toxic, severe toxic, obscene, threat, insult, identity hate.

5.2 EVALUATION OF SENTIMENT ANALYSIS

The Sentiment of the comment in each tuple of a feature set can be calculated using Bert model with pipeline and Naive Bayes classifier. Below table is a results of sentiments.

Table 5.1: ANALYSIS OF SENTIMENT SCORE

Techniques used	Sentiment analysis
BERT	99%
Naive Bayes classifier	76%

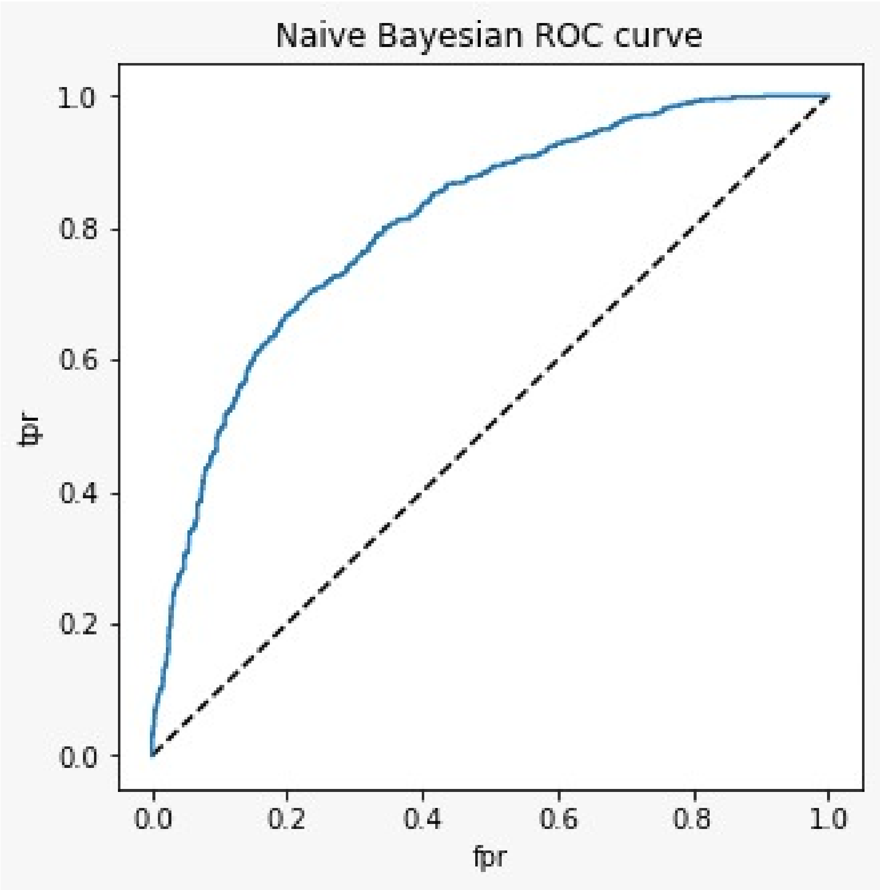


Figure 5.1: Naive Base Curve for Sentiment Analysis

Figure 5.1 shows the output curve of sentiment analysis using naive bayes classifier.

5.3 WORD CLOUD GENERATION FOR SENTIMENT ANALYSIS

Finally, the word cloud is generated for the words frequented in sentiment analysis as positive and negative. Figure 5.2 shows the words frequented in sentiment analysis

Figure 5.2: Words Frequented as Negative and Positive

5.4 ANALYSIS USING DEEP LEARNING

LSTM model managed to reach an impressive 81 percent validation accuracy on the 10th epoch and 95 percent at 20th epoch. Nonetheless, the over-fitting problem is still persistent in the model.

This could be further reduced by introducing a more aggressive

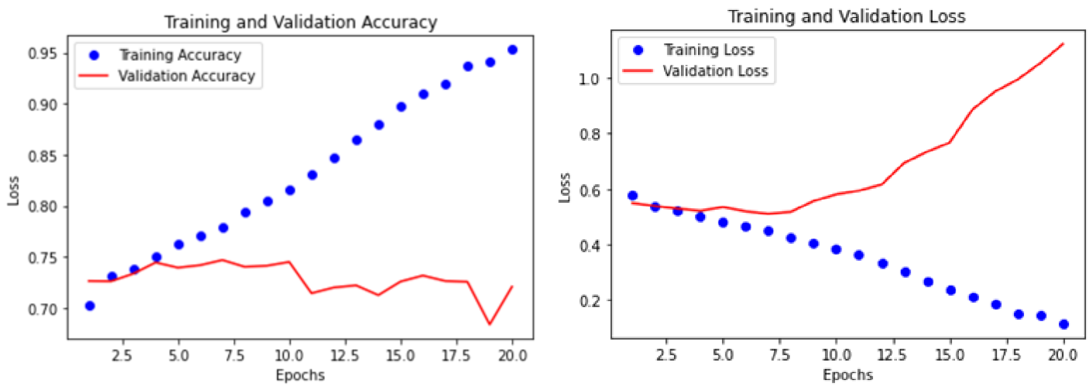


Figure 5.3: LSTM Accuracy Graphs

regularization and training the model for a much bigger number of epochs, and by also training the model on a bigger, more diverse, cleaner data. Regularization is the process of preventing a model from over-fitting the training data. Here, we have used regularization-dropout for the expected result. Figure

5.3 shows the LSTM accuracy graphs for LSTM model.

After applying regularization technique, the model accuracy has plateaued,

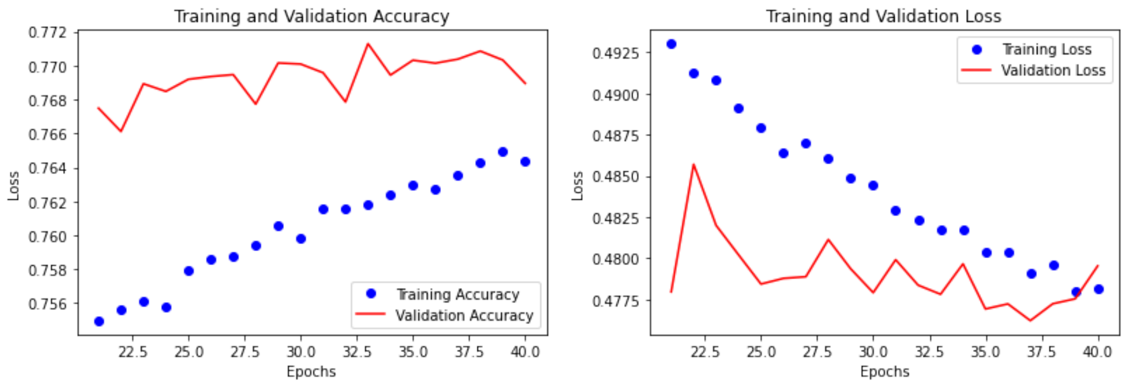


Figure 5.4: LSTM Model - Dropout Graphs

reaching its best validation value of 77.1 percent.we can conclude that the regularization process did not really help us in our case. A tiny 0.5 percent improvement was observed after adding dropout to the model. Figure 5.4 shows the dropout graphs of LSTM model

5.5 EVALUATION OF TOXICITY DETECTION

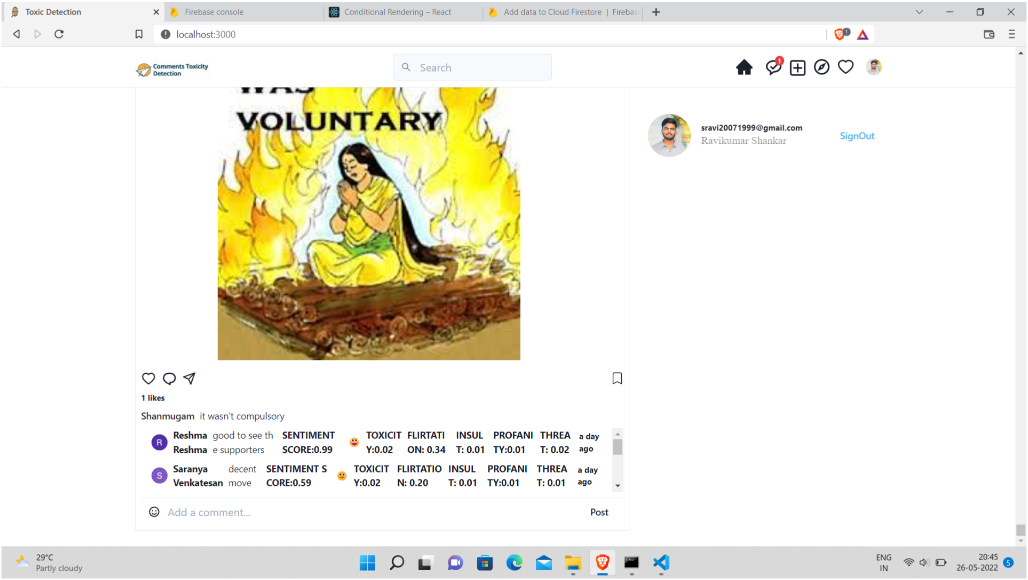
Toxicity of the comments in each tuple of a feature set can be calculated using Logistic Regression, KNeighbors Classifier, MultinomialNB, BernoulliNB, Linear SVC and Random Forest Classifier. Below table is a comparison of the results of toxicity of comments using Logistic Regression, KNeighbors Classifier, MultinomialNB, BernoulliNB, LinearSVC and Random Forest Classifier. Figure 5.5 shows the words frequented in sentiment analysis

	Log Regression	KNN	BernoulliNB	MultinomialNB	SVM	Random Forest
F1 Score(toxic)	0.861234	0.185120	0.776521	0.874958	0.876133	0.838055
F1 Score(severe_toxic)	0.927879	0.857416	0.803707	0.936170	0.926004	0.934874
F1 Score(obscene)	0.908655	0.519056	0.787830	0.901463	0.921378	0.909091
F1 Score(insult)	0.896599	0.257992	0.783762	0.897411	0.902619	0.883993
F1 Score(threat)	0.628821	0.720000	0.311828	0.504762	0.786765	0.795539
F1 Score(identity_hate)	0.699029	0.230159	0.549206	0.485857	0.797516	0.768448

Figure 5.5: Toxicity Detection using various Models

5.6 WEB APPLICATION

The layout and features of the web application (i.e, comments, post, likes, toxicity level and sentiment score) are highlighted in the below images.



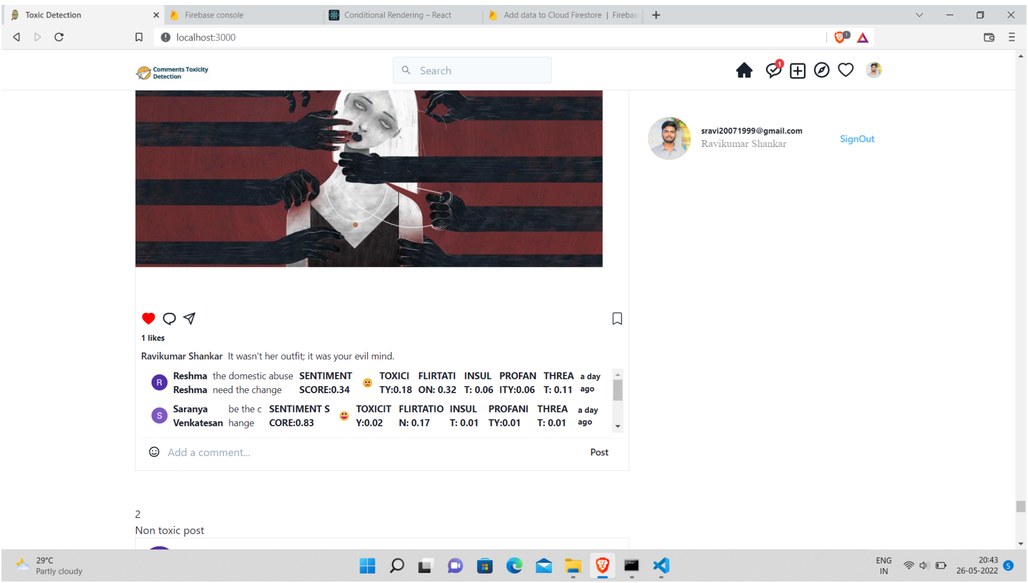


Figure 5.6: Layouts of Web App

Figure 5.6 shows the layout of the web application using the above mentioned functionalities.

5.7 DETECTING THE NATURE OF THE POST

The nature of the post (whether toxic or non-toxic) is mentioned at the top-left corner of the posts uploaded in the web application. The images of web application are mentioned below:

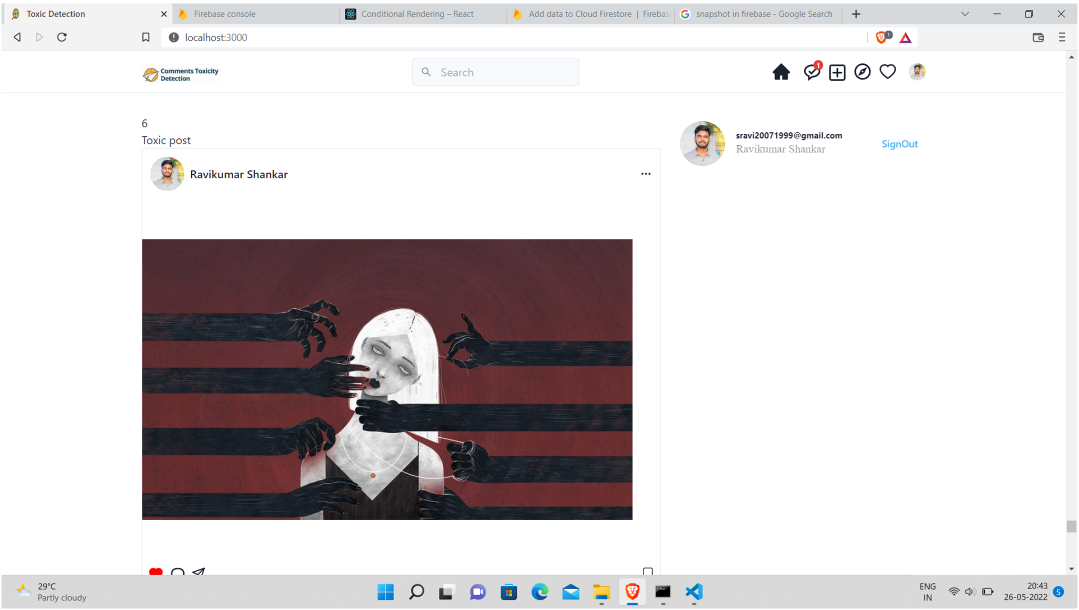


Figure 5.7: Toxic Post

Figure 5.7 shows the toxic nature of the post in the web application

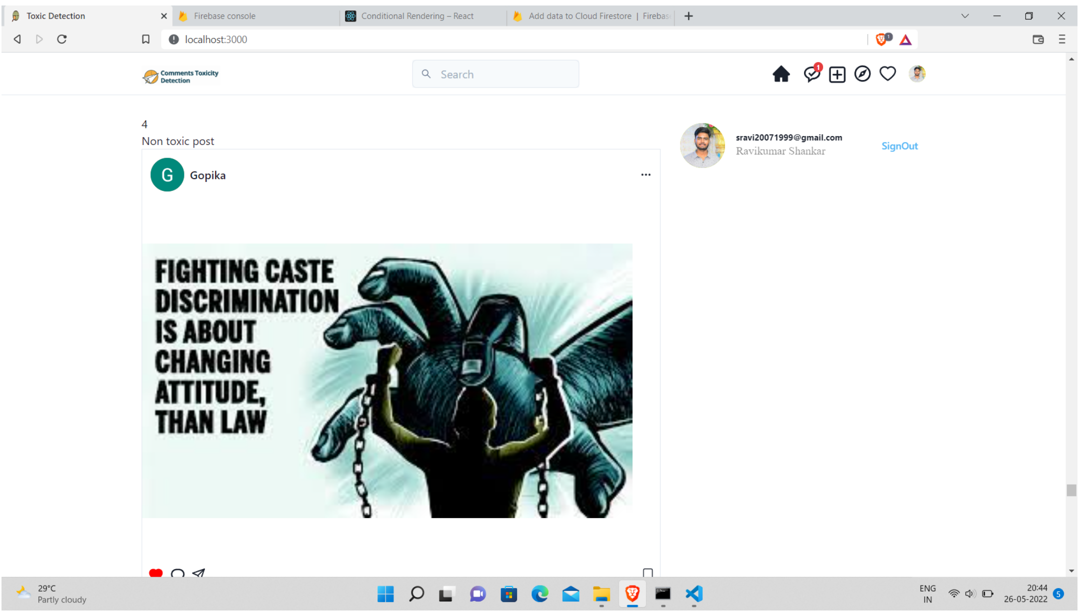


Figure 5.8: Non-toxic Post

Figure 5.7 shows the non-toxic nature of the post in the web application

CHAPTER 6
CONCLUSION AND FUTURE WORK

Analysing toxicity over user profiles using deep learning focuses on analyzing the sentiments and toxicity of the comments and feeding the data to a machine learning model to train it and then check its accuracy. We measured the performance of our classifiers by considering the data from the web application and we obtained a remarkable 92 percent accuracy for sentiment analysis from which it was possible to deduce that BERT's language modeling power significantly contributes to achieve a good text classification. The obtained toxicity score will be helpful in identifying fake accounts and the spread of negativity. Here, we obtained a remarkable 60 to 65 percent accuracy whether the conversation will become more toxic than expected and also predict whether the next reply posted by a specific user will be toxic, with an accuracy of up to 70.5 percent. Then with the help of threshold optimization iterative algorithm, the threshold frequency of 0.4 is obtained as default with which the user profiles are analysed and if the overall threshold frequency is below 0.4, then the users are acknowledged regarding the post.

The developed system proves to be a potential tool in real time as well as various fields of research and development. Further development in the existing system can be done in order to improve the accuracy of the system. The future work will be mainly focusing on upgrading existing system and developing it into a mobile app. Additional features such as sending notification through the verified mail id and giving accessablity to login with furthermore options such as facebook, twitter, etc are more efficient to make it a social project. This process mainly focuses on the background of the user such as their social reputation (maximum number of followers), global engagement, topic engagement, likability.

REFERENCES

[1] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova, *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*, 2019.

[2] K. Dashtipour, M. Gogate, A. Adeel, A. Algarafi, N. Howard and A. Hussain, *"Persian Named Entity Recognition", 2017 IEEE 16th International Conference on Cognitive Informatics Cognitive Computing (ICCI*CC)*, pp. 79-83, 2017.

[3] Xin Li, Lidong Bing, Wenxuan Zhang, and Wai Lam. 2019.*Exploiting BERT forend-to-end aspect-based sentiment analysis*. arXiv preprint arXiv:1910.00883(2019).

[4] Aditi Gupta, Hemank Lamba, Ponnurangam Kumaraguru, "1.00 per RT BostonMarathon, PrayForBoston: Analyzing Fake Content on Twitter", In the proceedings of *Indraprasatha Institute of Information Technology*, Delhi, India, pp 1-12, September 2013.

[5] Agarwal A, Xie B, Vovsha I, Rambow O, Passonneau RJ (2011) Sentiment analysis of twitter data. In: Proceedings of *the workshop on language in social media* (LSM 2011), pp 30–38

[6] Ahuja S, Dubey G (2017) Clustering and sentiment analysis on twitter data. In: *2017 2nd international conference on telecommunication and networks (TEL-NET)*, IEEE, pp 1–5

[7] B. Patel and D. Shah, "Significance of Stop Word Elimination in Meta Search Engine", Proceedings of *the 2013 International Conference on Intelligent Systems and Signal Processing (ISSP)*, Mar 1–2, 2013.

[8] Honglei Liu; Yan Xu; Zhiqiang Zhang; Ni Wang; Yanqun Huang; Yanjun Hu; Zhenghan Yang; Rui Jiang; Hui Chen, *A Natural Language Processing Pipeline of Chinese Free-Text Radiology Reports for Liver Cancer Diagnosis*,28 August 2020

[9] Fabian Riquelme, Pablo Gonzalez-Cantergiani, ”Measuring User Influence on Twitter: A survey”, *Journal of Information Processing and Management*, vol. 52, issue. no. 5, pp 949–975, April 2016.

[10] Yu L., Wang J., Lai K.R., Zhang X. *Refining Word Embeddings Using Intensity Scores for Sentiment Analysis*. IEEE ACM Trans. Audio Speech Lang. Process. 2018

[11] Hutto C J, Sarita Yardi, Eric Gilbert, ”A Longitudinal Study of Follow Predictors on Twitter”, In the proceedings of the *SIGCHI Conference on Human Factors in Computing Systems (CHI ’13)*, Association for Computing Machinery, New York, NY, USA, pp 821–830, April 2013.

[12] Isabel Anger, Christian Kittl, ”Measuring Influence on Twitter”, In the Proceedings of the*11th International Conference on Knowledge Management and Knowledge Technologies*, Graz, Austria, pp 1-4, Sep 2011.

[13] Wei, Wei, Cong, Gao, Miao, Chunyan, Zhu, Feida, Li, Guohui , ”Learning to Find Topic Experts in Twitter via Different Relations”, *IEEE Transactions on Knowledge and Data Engineering*, vol. 28, issue. no. 7, pp 1764-1778, July 2016.

[14] WENG, Jianshu, LIM, Ee Peng, JIANG, Jing, HE, Qi, ”Twitterrank: Finding Topic-Sensitive Influential Twitterers”, In the proceedings of the *Third ACM International Conference on Web Search Data Mining*, New York, pp 261-270, February 3-6, 2010.

[15] Zhang X, Han D-D, Yang R, Zhang Z, ”Users Participation and Social Influence During Information Spreading on Twitter”, *PLOS ONE*, vol. 12, issue. no. 9, Sep 2017.

[16] C. S. Perone, ”Machine learning: : Text feature extraction (tf-idt) - Part 1”, pyevolve.sourceforge.net, September 2011.

[17] T. Mikolov, K. Chen, G. Corrado and J. Dean, *”Efficient Estimation of Word Representations in Vector Space”*, arxiv.org., September 2013.

[18] J. Wu and L. Liu, *”Kernel-Based Method for Automated Walking Patterns Recognition Using Kinematics Data”, 5 th Workshop on Natural Language Processing* , 2006.

[19] Pooja Kumari, Shikha Singh, Devika More and Dakshata Talpade, *”Sentiment Analysis of Tweets”, IJSTE - International Journal of Science Technology Engineering*, vol. 1, no. 10, pp. 130-134, 2015, ISSN 2349-784X.

[20] Rasika Wagh and Payal Punde, *”Survey on Sentiment Analysis using Twitter Dataset”, 2nd International conference on Electronics Communication and Aerospace Technology (ICECA 2018) IEEE Conference*, ISBN 978-1-5386-0965-1,2018.