A

Project Report

on

Toxic Comment Detection and Classification

SUBMITTED TOWARDS THE
FULFILLMENT OF THE REQUIREMENTS OF

Bachelor of Engineering (Computer Engineering)

BY

Ajay Kumar	Exam No: B150224206
Kumar Shivam	Exam No: B150224253
Naresh Kumar	Exam No: B150224269
Rahul Malhan	Exam No: B150224275

Under The Guidance of

Prof. Sharayu Lokhande

Department of Computer Engineering

Army Institute of Technology, Pune - 411015.

SAVITRIBAI PHULE PUNE UNIVERSITY

2020-21

ARMY INSTITUTE OF TECHNOLOGY,
DEPARTMENT OF COMPUTER ENGINEERING

CERTIFICATE

This is to certify that the Project Entitled

Toxic Comment Detection and Classification

Submitted by

Ajay Kumar	Exam No: B150224206
Kumar Shivam	Exam No: B150224253
Naresh Kumar	Exam No: B150224269
Rahul Malhan	Exam No: B150224275

is a bonafide work carried out by students under the supervision of Prof. Sharayu Lokhande and it is submitted towards the fulfillment of the requirement of bachelor of engineering (Computer Engineering) Project.

Prof. Sharayu Lokhande
Internal Guide

Prof.(Dr.) S. R. Dhore
H.O.D

_____
External Examiner

Dr. B P Patil
Principal

Place : AIT, Pune

Date :

PROJECT APPROVAL SHEET

A

Project Stage-II Report

on

Toxic Comment Detection and Classification

is successfully completed by

Ajay Kumar	Exam No: B150224206
Kumar Shivam	Exam No: B150224253
Naresh Kumar	Exam No: B150224269
Rahul Malhan	Exam No: B150224275

at

Department Of Computer Engineering

Army Institute of Technology, Pune-411 015.

SAVITRIBAI PHULE PUNE UNIVERSITY

2020-21

Prof. Sharayu Lokhande Department of Computer Engg.	Prof.(Dr.) S R Dhore Head

INDEX

Abstract

Toxic comments refers to hatred online comments classified as disrespectful or abusive towards individual or community. With a boom of internet, lot of users are brought to online social discussion platforms. These platforms are created to exchange ideas, learning new things and have meaningful conversations. But due to toxic comments many users are not able to put their points in online discussions. This degrades quality of discussion. In this paper we will check the toxicity of comment. And if the comment is toxic then classify the comments into different categories to examine the type of toxicity. We will utilize different machine learning and deep learning algorithms on our dataset and select the best algorithms based on our evaluation methodology. Moving forward we seek to attain high performance through our machine learning and deep learning models which will help in limiting the toxicity present on various discussion sites.

Keywords : Toxic Comments, Natural Language Processing, Machine Learning, Deep Learning, Text Classification, Multilabel Classification

Acknowledgments

It gives us great pleasure in presenting the preliminary project report on ***'Multilabel Toxic Comment Detection and Classification'***.

We would like to take this opportunity to thank my internal guide **Prof. Sharayu Lokhande.**

Prof.(Dr.) S R Dhore

Dr. B P Patil

Ajay Kumar
Kumar Shivam
Naresh Kumar
Rahul Malhan
(B.E. Computer Engg.)

List of Figures

Chapter 1

INTRODUCTION

There is increase in number of people using internet. Internet is main invention for 21st century. According to website , the number of internet users have increased from 1100 million in 2005 to 3969 million users in 2019 which is staggering 260% increase [1]. Hence, more people are using social networking and online discussion platforms.

There is also a huge shift the way internet is used. In the initial days of the internet, people used to communicate with each other through Email. But with a platform like social media, we see that people got a way to keep in touch with their long-lasting friends, meet new peoples having same interests and hobbies. We are now more connected than ever. Not only discussion of friends and people, but social media has also evolved to support business needs.

With increase in services like gaming and live streaming, more velocity of comments is added to sites. Social media has broken down many of the communication barriers between different consumer groups as well as between individuals. Hence no doubt that social media sites such as Facebook, Twitter, Reddit, etc. have become billion-dollar companies.

Over these all years we have seen lot of instances where social media have played pivot role due to toxic comments and hatred. For example, Chief Minister of Uttar Pradesh State of India blamed social media like Facebook, Twitter, and YouTube for escalating tensions during communal conflict between Hindu and Muslim community in Muzzafarnagar, India in 2013 [2]. Kalamboli police on booked a man for abusing and threatening the police via a comment on a Facebook post [3]. Another example is of Riots that took place in DJ Halli, Bengaluru, India in 2020

over a provocative Facebook post against Islam that left 3 dead and many injured [4].

On January 6, 2021 US Capitol Riots took place by supporter of Donald Trump. Many extremists had posted on Social Networking sites posts such as “occupy the Capitol”, “bring revolution”, etc. before riots [5]. Hence, it is very important to detect such threats, hatred, toxicity on online discussion platforms and social networking sites. Because not doing so can cause violence, riots, prevent good debates, make internet an unsafe place and can affect people mentally.

Let us take an example of comment present in our dataset “Just shut up and stay shut. Don’t edit anymore”, it can be easily identified that the phrases like “shut up”, “Don’t edit anymore”, etc. are negative and thus this comment is toxic. But it besides toxic we need to go through series of steps to classify comment using machine learning classification algorithms to verify type of toxicity of obtained results.

We will use different machine learning and deep learning models on our Data set which is made available by Conventional AI in Kaggle.com. In this paper we will use Logistic Regression and Support Vector Machine Models with TF-IDF Vectorizer, Long Short-Term Memory with Glove and Word2Vec Embedding. We have used all models on given dataset and compare their scores to find which one will be best.

There is lot of information being delivered every time on social media sites. There is increase in hate speech that both may promotes violence towards Muslims and Arabs after following extremist violence events [6]. Because of this negativity, particular community might feel insecurity while utilizing social platforms. There was survey conducted asking American adult about problem of online harassment or bullying. Roughly four-in-ten Americans have personally experienced online harassment. 62% percent of participant in study considered it as major problem whereas 33% considered as minor problem. A total of 95% called it problem and 35% agreed for online companies to build better policies and tools for their platforms [7]. Due to negativity, civilized conversations via social media are not present since hateful remarks are limiting individual to communicate and to have contradicting feelings [8].

There were endeavours by people to build the online wellbeing by moderating websites through crowdsourcing schemes and remark criticizing, much of the time these procedures neglect to recognize the toxicity [9]. Along these lines, we need to track down a potential method that can recognize the online toxicity of client content successfully [10].

As Computer understands binary information and in real world we have information in different structures for example pictures or text. So, we need to change over the information of real world into binary for legitimate processing through the computer. In this paper, they have utilized this changed over information and apply Machine learning strategies to arrange online remarks [11].

Nguyen and Nguyen [12] made model consisting of 2 components – Deep Learning Classifier and Tweet Processor. Tweet Processor is used for applying semantic rules and preprocessing on datasets to capture important information. They used character-level embeddings to increase information for word-level embedding. They then used DeepCNN for character level embeddings. After that a Bidirectional Long Short-Term Memory network (BiLSTM) produces a sentence-wide feature representation from the global fixed size feature and wordlevel embedding. Their model produce an accuracy of 86.63% on Standford Twitter Sentiment Corpus.

Liang-Chih Yu et al. [13] proposes a word vector refinement model. This refinement can be applied to any of already trained word vectors. Based on semicolon lexicon, their model gives high rank to sentimentally similar neighbour and vice versa. Their experimental results show that their proposed method can perform better for various neural networks. It also have better performance for both sentimental embedding and conventional word embeddings.

A method was introduced by Wulczyn et al. [14] develop method to analyze personal attacks. They generated over 63M machine labeled and 100k human labeled comments. They found that attacks on Wikipedia are not limited to a set of users.

Hossein Hosseini et al. [15] apply the attack on the Perspective toxic comment detection website. This website gives toxic score to any phrase. They tried to modify a toxic phrase having same meaning so that model will give it very low toxic scores. This existence is harmful for toxic detection system. In another method-

ology, Convolutional Neural Networks (CNN) was utilized in text characterization over online substance [16], with no information on syntactic or semantic language.

Y. Chen et al. [17] propose the Lexical Syntactic Feature (LSF) architecture to distinguish offensive content and recognize likely offensive clients in web-based media. Their experiments shows that LSF algorithms for sentence and user offensiveness outperformed traditional learningbased methods. Their LSF can adapt to various writing styles of English language and can tolerate informal and misspelled content.

Jigsaw and Google's Counter Abuse Technology team introduce project named Perspective. It uses machine learning models to identify abusive comments. The models score a phrase based on the perceived impact the text may have in a conversation and have capability to classifying comments.

Navoneel Chakrabarty [18] utilizes machine learning model on Jigsaw Toxic Comment Detection dataset to label toxicity of comments and produce the Mean Validation Accuracy, so obtained, is 98.08

Chapter 2

ANALYSIS

2.1 Project Plan

2.1.1 Project Tasks

Major Tasks in the Project stages are:

- Task 1: Study and analysis of existing machine learning and deep learning models for toxic comment detection and classification.
- Task 2: Finding and gathering dateset available on kaggle.
- Task 3: Data Preprocessing and making Embedding using Glove and Word2Vec Embeddings.
- Task 4: Checking which machine or deep learning model is best.

2.2 Requirment Analysis

2.2.1 HARDWARE AND SOFTWARE REQUIREMENTS

- Python Platform (Anaconda, Spyder, Jupyter)
- NLTK package, other machine learning related packages
- Modern Web Browser
- Kaggle Dataset

- Intel(R) Core (TM) i5-8250U processor, CPU @ 1.60 GHz and 8 GB RAM, GPU NVIDIA 940MX by Python's Scikit-Learn Machine Learning Toolbox or utilize Google Colab

2.2.2 Function Requirements

- Collect comments in a real time fashion. For current training we utilized Kaggle Dataset – Toxic Comment Detection and Classification
- Store data in database and pre-process data to compensate for missing values, feature scaling, etc.
- Train machine learning model which will predict the comment is toxic or not. (Classification Problem).

2.2.3 Non-Function Requirements

- Should be able to give desired result of comment being toxic or not within short span of life
- Should predict accurate information
- Should be able to determine type of toxicity

Chapter 3

DESIGN

3.1 Architectural Design

A description of the program architecture is presented. Subsystem design or Block diagram, Package Diagram,Deployment diagram with description is to be presented.

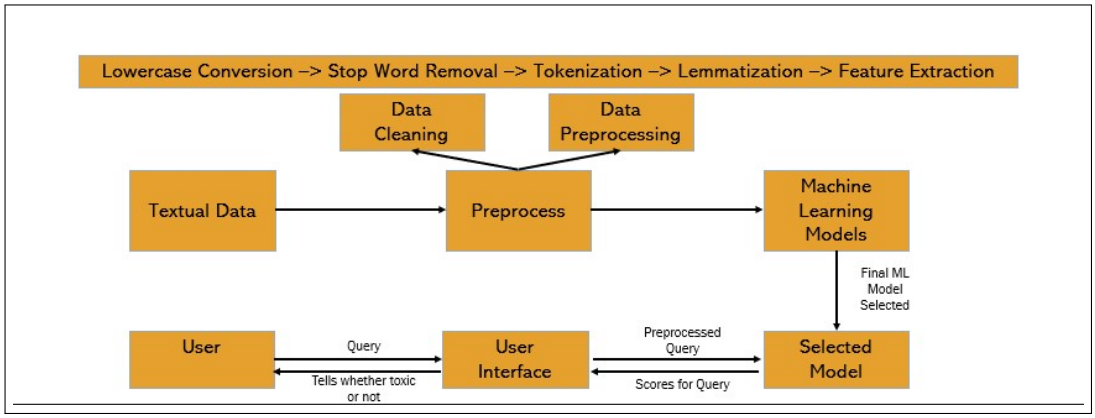

Figure 3.1: Architecture diagram

3.2 Software Requirement Specification

1. Introduction

   (a) Purpose

   The project aims at developing machine learning system for detection of toxic comments in social media and various blogs.

   (b) Scope

   Due to the exponential development of computer science and technology

provides us with one of the greatest innovations of the ”Internet” of the 21st century, where one person can communicate to another worldwide with the help of a mere smartphone and internet. With the advancement of social media, it becomes highly important to classify the content into positive and negative terms, to prevent any form of harm to society and to control antisocial behaviour of people. By building machine learning model which effectively detects and can classify comments on various online forums, social media sites, we can prevent violence, riots and even identify hate speeches. The feature can also be used to prevent posting of comments beforehand on various social media sites thus by reducing major chunk of work need to be done by companies/authorities.

2. Overall Description

   (a) Product Perspective

   The product is independent and totally self-contained. We will be making machine learning model and make a simple website with text box where user can type comments/statements and get response whether given comment is toxic or not and if toxic which category it belongs.

   (b) User Characteristics

   General characteristics of the intended users of the product are Basic knowledge of working with Software. There is no technical knowledge required for working with software. Only requirement is that user should be provided web portal which he must have knowledge of.

   - Regulatory Policies:

     The data we use is from Kaggle and this data is accessible by trustworthy high-ranking government officials and after processing and encrypting given to the companies.

   - Hardware Limitation:

     When this project is implemented on large scale, we will need lots of high-performance GPU/CPU to train model on this large data.

   (c) Assumptions and Dependencies

We are assuming to have infrastructure for processing capability to run Machine Learning Prediction, so that raw image data can be processed and send it to server as textual data.

3. Specific Requirements

   (a) Function Requirement

   - Collect comments in a real time fashion. For current training we utilized Kaggle Dataset – Toxic Comment Detection and Classification
   - Store data in database and pre-process data to compensate for missing values, feature scaling, etc.
   - Train machine learning model which will predict the comment is toxic or not. (Classification Problem).
   - Train machine learning model which will allow users to select which types of toxicity they are interested in finding.

   (b) Performance Requirements

   - Should be able to give desired result of comment being toxic or not within short span of life
   - Should predict accurate information.

   (c) System Requirements/ Developer Requirement

   - Python Platform (Anaconda, Spyder, Jupyter)
   - NLTK package, other machine learning related packages
   - Modern Web Browser
   - Kaggle Dataset
   - Intel(R) Core (TM) i5-8250U processor, CPU @ 1.60 GHz and 8 GB RAM, GPU NVIDIA 940MX by Python's Scikit-Learn Machine Learning Toolbox or utilize Google Colab.

   (d) Reliability

Reliability depends on widespread data available, processing power, accuracy of machine learning models.

(e) Maintainability

Machine learning model need to be updated with increase in dataset as per availability.

3.3 Risk Management

3.3.1 Risk Identification

Risks are identified from different phases of the project development from requirements and the SRS Document. The following three risks were identified:

- Technical Risk.
- Operational Risk.
- Schedule Risk.

3.3.2 Risk Analysis

The risks for the project are analyzed within the constraints of time and quality.
Technical Risks

- Overfitting and Underfitting.
- Biased dataset.
- Poor problem-solution alignment.

Risk Solution

- Cross validation.
- Working on genuine dataset collected by renowned institute.
- Ensembling.

Operational Risks

- Difficulties in operating on huge datasets (100GB).
- Resources are limited in open-source platforms (e.g., Google Colab).
- Dependency on cloud services.
- Difficulties in satisfying large demand.
- Technology Troubles
- Security snags
- Models Misbehaving
- Interaction Issues

Risk Solution

- For operating on large datasets, we need to shift to expensive virtual machines.
- Deploy model on paid reliable services.
- To fulfill large demand, we can host multiple servers in backend.

Schedule Risks

- The Phase wise plan may get delayed due to some unavoidable circumstances.
- Feature selection and epoch size during testing may consume time.
- The project completion may get delayed if the schedule is not followed.

Chapter 4

MODELING

4.1 UML Diagrams

4.1.1 Sequence Diagram

The sequence diagram simply depicts the interaction between the objects in sequential order, that is, the order in which these interactions take place. We can also use the terms event scenarios or event diagrams to refer to a sequence diagram. Sequence diagrams describe how and in what arrangement of objects in the system function.

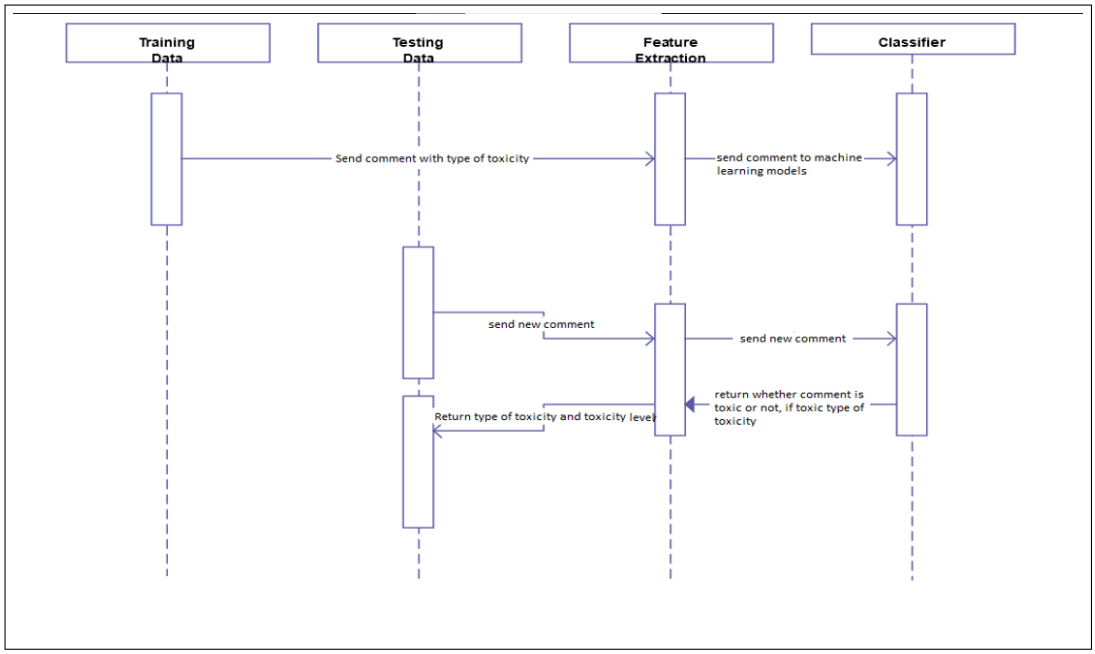

Figure 4.1: Sequence diagram

4.1.2 Use Case Diagram

A use case diagram in its simplest form is a representation of the user's interaction with the system that illustrates the relationship between the user and the different use cases that the user engages in. A use case diagram can define different types of system users and different use cases and is often accompanied by other types of diagrams as well. Use cases are either represented by circles or xellipses.

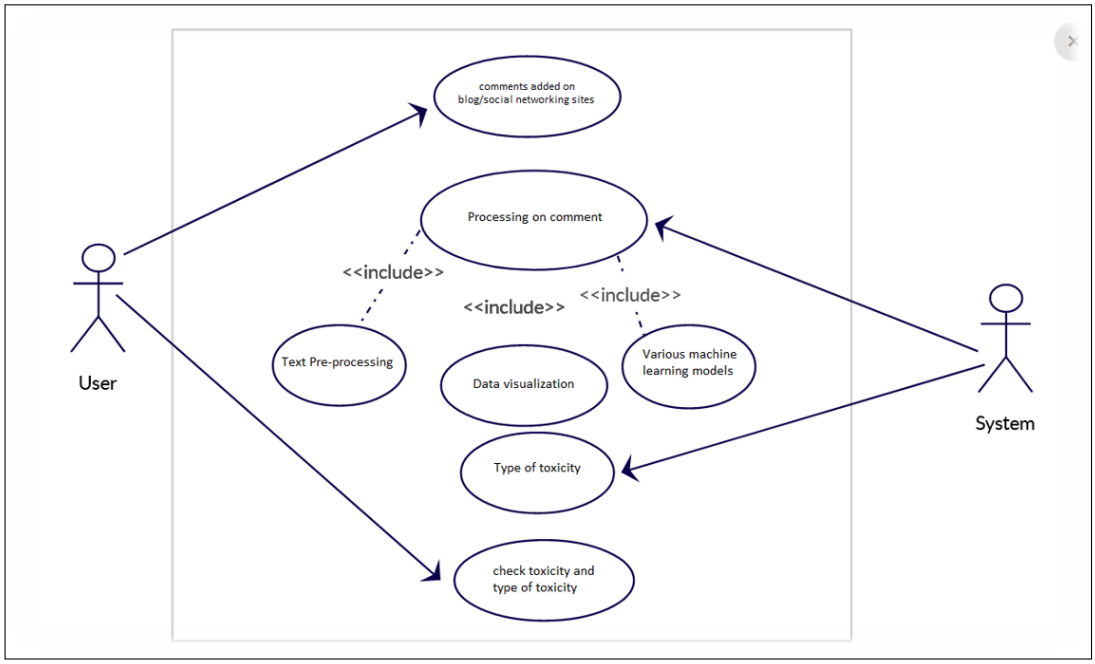

Figure 4.2: Use Case diagram

4.1.3 Activity Diagram

An activity diagram is a behavioral diagram i.e. it depicts the behavior of a system. An activity diagram portrays the control flow from a start point to a finish point showing the various decision paths that exist while the activity is being executed.

4.1.4 Component Diagram

A component diagram breaks down the actual system under development into various high levels of functionality. Each component is responsible for one clear aim

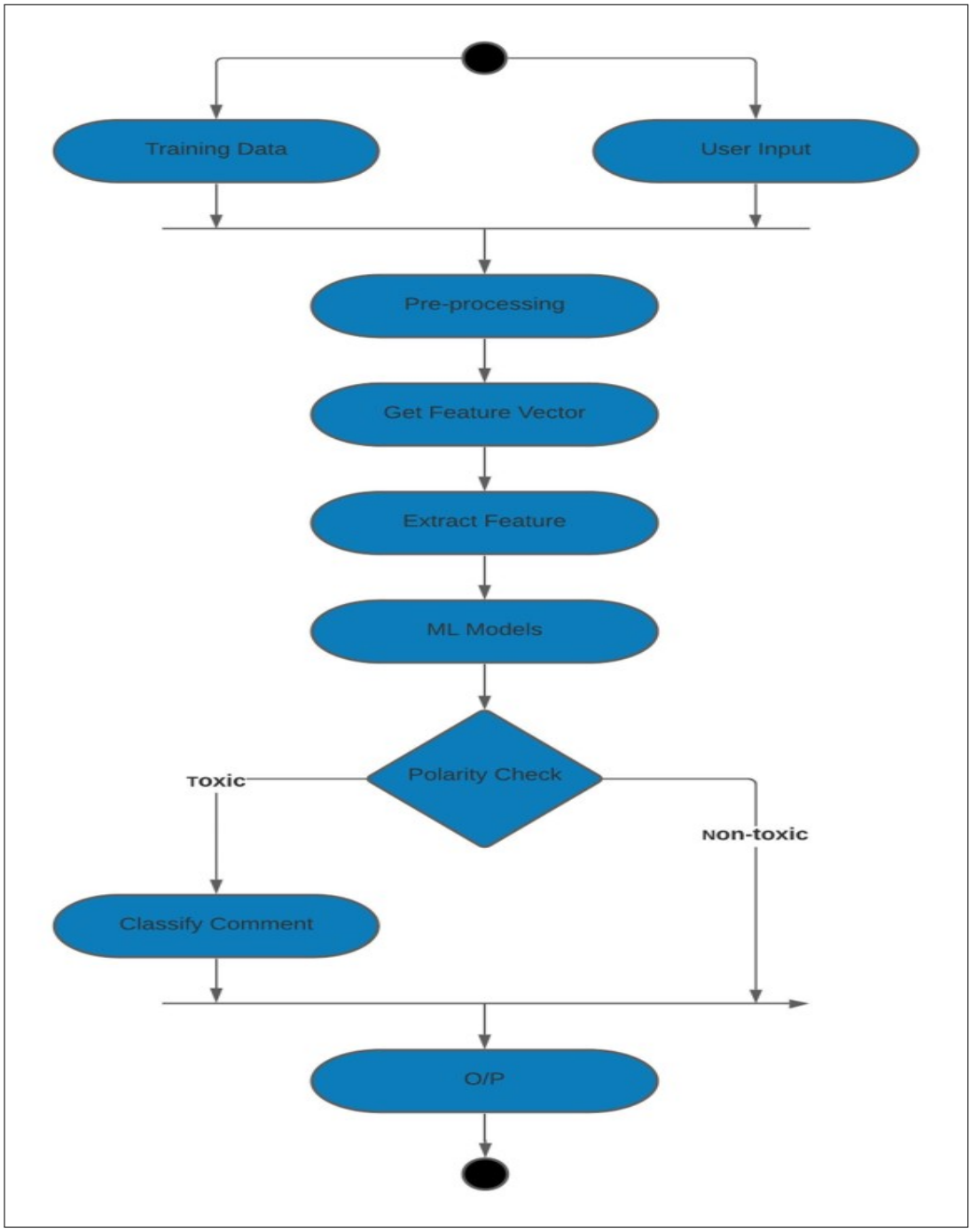

Figure 4.3: Activity diagram

within the entire system and only interacts with other essential elements on a need-to-know basis.

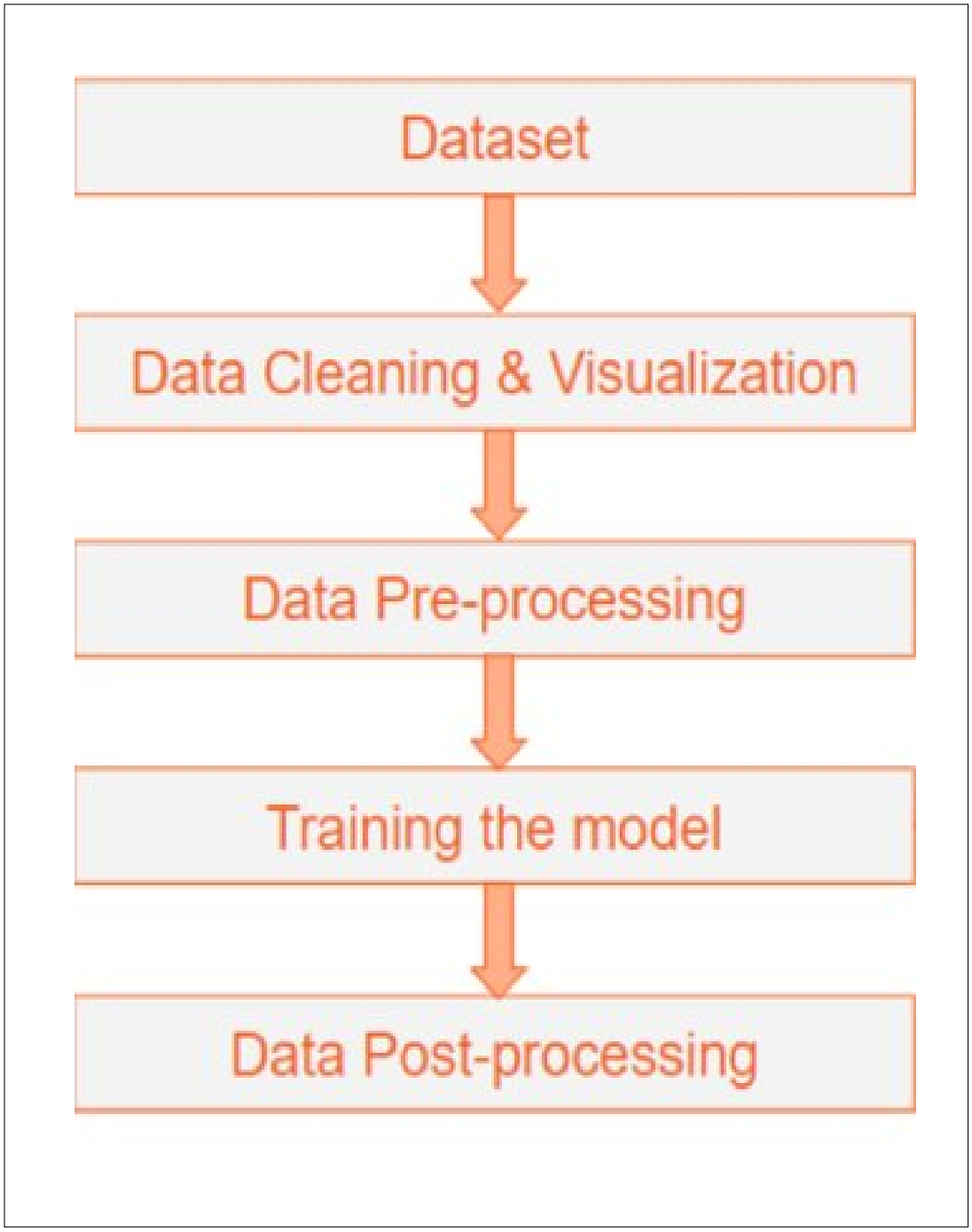

Figure 4.4: Component diagram

Chapter 5

CODING

We have taken dataset provide in Kaggle website provided by Conventional AI. It is collection of large number of comments in Wikipedia website and labeled as - toxic, severe toxic, threat, identity hate, obscene and insult. The advantage of this type of data is that these comments represent a true sample of the content present on the social media sites. We first ran analysis and visualization on this data. For our Machine Learning model, we have removed outliers and noise which is present is data. We initially tested the performance of classical models namely, Support Vector Machine and Logistic Regression on this task with TF-IDF Vectorizer. We then applied pretrained embeddings, namely GloVe and Word2Vec in our model and performed the classification. We compared the performance of all the models using the mean AUC ROC score, Hamming and Log loss as the performance metric.

	id	comment_text	toxic	severe_toxic	obscene	threat	insult	identity_hate
0	0000997932d777bf	Explanation\nWhy the edits made under my usern...	0	0	0	0	0	0
1	000103f0d9cfb60f	D'aww! He matches this background colour I'm s...	0	0	0	0	0	0
2	000113f07ec002fd	Hey man, I'm really not trying to edit war. It...	0	0	0	0	0	0
3	0001b41b1c6bb37e	"\nMore\nI can't make any real suggestions on ...	0	0	0	0	0	0
4	0001d958c54c6e35	You, sir, are my hero. Any chance you remember...	0	0	0	0	0	0

Figure 5.1: Dataset Head

5.1 Algorithms/Flowcharts

The preprocessed data is still in natural language form, which is not understandable by the machine. So, it needs to be broken down into tokens or words, and these tokens need to be encoded in the form of integer or floating point numbers, to be used by machine learning algorithms for prediction. This complete process of converting the preprocessed text in natural language form into encoded form is called vectorization or feature extraction.

5.1.1 Count Vectorizer

This vectorizer encodes the text by scanning through every comment text and building a vocabulary of words, containing the words from all the comments. Then, it finds the count of words present in the vocabulary, for each comment. Finally, it creates a vector of dimension $|V||D|$ representing the encoded form of the comment texts, where $|V|$ denotes the number of words in the vocabulary, and $|D|$ denotes the number of comment texts i.e. the number of documents. Though this is the simplest technique to encode text, it suffers from drawbacks. It finds the count of a term locally in a comment, and does not consider the count of the term in the whole list of comments. The words that occur in most of the documents are not much important, therefore they must be penalized, which is not done by Count Vectorizer. This problem is solved by the TF-IDF Vectorizer.

5.1.2 TF IDF Vectorizer

This vectorizer also normalizes the text, i.e. reduces the impact of the tokens occurring in most of the documents, apart from tokenizing and counting. The value of the TF-IDF term in the vector increases if the frequency of the term is more in the comment text. However, the TF-IDF term decreases if the token appears in most of the comments. TF-IDF stands for Term Frequency Inverse Document Frequency. It consists of two parameters - term frequency (TF) and inverse document frequency (IDF), which decide the final value of the TF-IDF term.

- **Term Frequency - TF (t, d):** This is a local parameter which calculates the frequency of a term 't' in a document 'd'. This is similar to the Count Vectorizer method of encoding text.

- **Inverse Document Frequency - IDF (t):** The IDF gives the inverse of the document frequency, i.e. it is a global parameter. Here, Document Frequency DF (t) is the count of documents (i.e. comments) that contain a tern t 't'. Therefore, it calculates the number of documents in which a term appears and it takes the inverse and log of that. In order to avoid a zero-division error, an extra 1 is added to the denominator.

  $$IDF(t) = \log(\frac{1+|D|}{1+df(t)}) + 1$$

  where df(t) denotes the number of documents containing the term 't' and $|D|$ denotes the total number of documents.

The final TF-IDF term is calculated as follows:

$$TFIDF = TF(t,d)IDF(t)$$

In this formula, the first term increases if the word is frequent in the document. However, if the word appears in most of the documents, then the second term decreases. Therefore, the TF-IDF term considers both local and global impact of a word in a comment, and it penalizes those words that have a high document frequency and are not meaningful in making predictions. Hence, TF-IDF vectorizer performs better than Count Vectorizer for feature extraction.

5.1.3 Glove Embedding

Word embeddings are used to represent words in structured format. Because most of the data in internet is not structured, word embeddings techniques are a useful tool to transform data into more structured format so that useful information can be extracted.

In Bag of words models feature extraction can be done but they fail to capture any semantic or contextual information in texts. One-hot encoded vectors may lead to a highly sparse structure which causes the model to overfit. To overcome these shortcomings of the above approach, word embeddings are used.

Word embeddings represent words in the form of vectors in pre-defined dense vector space. These vectors contain meaningful semantic information about the words. The idea in this approach is words with similar semantic information are like each other. Hence, the similar words will be in proximity within the high dimensional vector space. Hence, we can significantly reduce the vector size in contrast to the one-encoding technique.

Pretrained word embeddings are obtained by unsupervised training of a model on a large corpus. As they are trained on a large corpus, they capture the semantic information of most of the words. These pretrained embeddings are provided by different companies and organizations for open use.

GloVe stands for Global Vectors. It is provided by Stanford as an open-source project. In this approach, a word co-occurrence matrix is constructed. This helps in capturing the semantic information. The co-occurrence matrix stores information about the frequency that appear in some context. Thus, it takes both local statistics and global statistics into account to obtain the embeddings.

5.1.4 Word2Vec Embedding

It is one of the earliest pretrained embedding. It has 2 flavors. First is Skip-Gram Model where the algorithm tries to predict the context or surrounding words in which the word would have been used. It learns by predicting the surrounding words given a current word. Second is Continuous Bag of Words (CBOW) model where the algorithm tries to predict the word if a context is given. In this way, the word embeddings vectors are generated.

5.1.5 Logistic Regression

Logistic Regression is the fitting regression analysis to use when the reliant variable has a binary answer. Like any remaining kinds of regression frameworks, Logistic Regression is likewise a predictive regression framework. Logistic Regression is utilized to assess the connection between one reliant variable and one or more non-reliant variable. It gives discrete yields going somewhere in the range of 0 and 1. Logistic Regression utilizes a more complex cost function; this cost function is

known as the 'Sigmoid function' or otherwise called the 'logistic function'.

$$f(x) = 1/(1+e^{-x})$$

5.1.6 Support Vector Machine

Support Vector Machines use the concept of support vectors and hyperplanes for classification tasks. They can be used for both regression and classification tasks but are more popular for classification. They can be used for both linear and non-linear data. It works by constructing an optimal hyperplane in an N-dimensional space i.e., a boundary separating the data points, such that the margins between the support vectors is maximized. Support vectors refers to the data points which are closest to the hyperplane and are useful for training tasks, and the margin refers to the distance between the two parallel lines passing through the closest support vectors on either side of the hyperplane.

In case of non-linearly separable data, it transforms the original data into higher dimensions for classification task. It is also known that SVM perform excellent for higher dimensional data because complexity of SVM does not depend on dimensionality of data used but on number of support vectors. This also helps it to be memory efficient.

In our case, we have used Support Vector Machine with Binary Relevance and Classifier Chains for predictions.

In Binary Relevance method, we transform the problem into separate single-class classification problems, each of the problems having a single label. We then apply the Support Vector Machine on each problem separately to get the result. After that, the results of each problems can be combined to get all the labels for a comment. Simple it is it comes with drawbacks. It ignores any correlation between the labels. Hence it will give poor results if there is correlation between labels.

In Classifier Chain method, we transform the problem into separate single-label classification problems, such that if ith classifier is trained on input variable(s) X, then (i+1)th classifier is trained on input variable X and output produced by ith

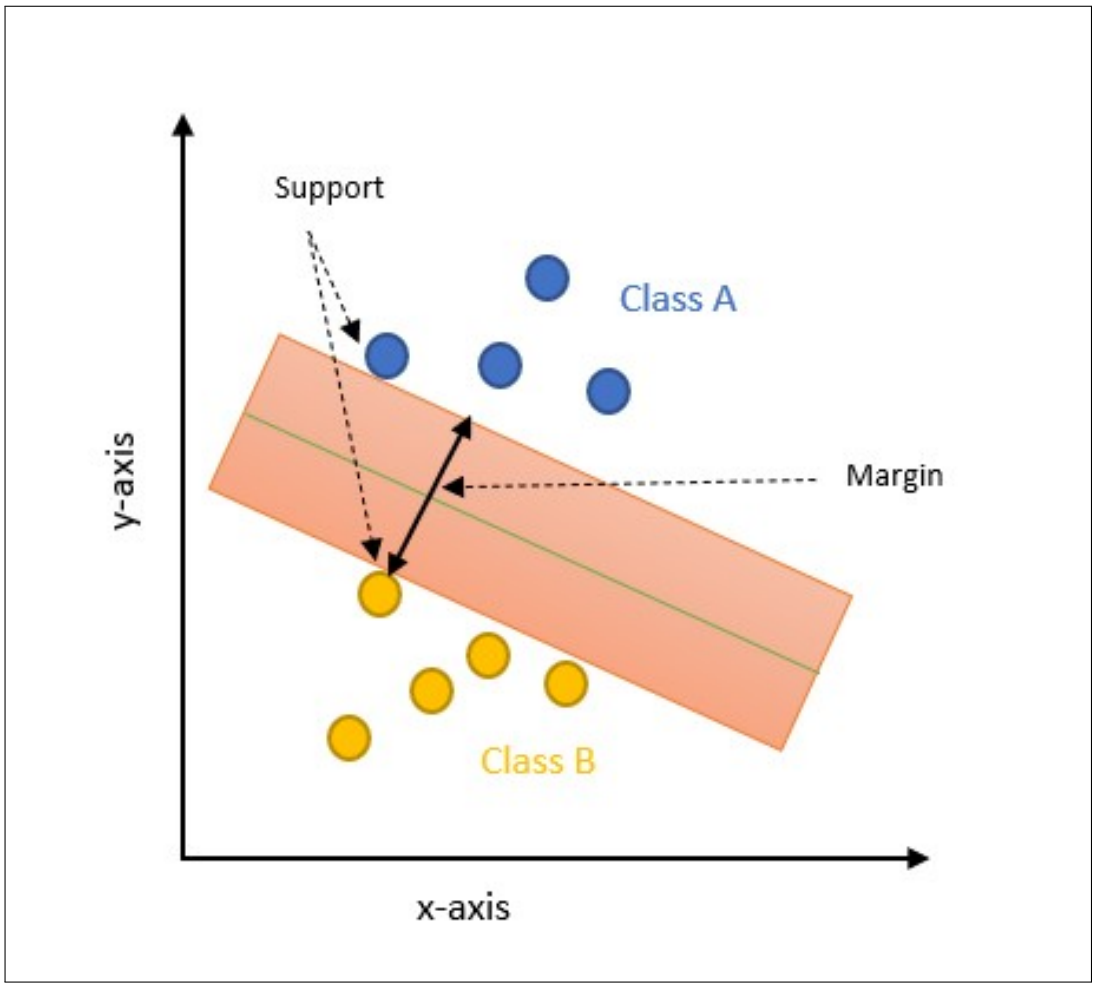

Figure 5.2: SVM

classifier. Hence, this technique considers the correlation between the labels, since for every new classifier, the predictions of the previous classifiers are considered, i.e., for a given target variable, it also considers the correlation between previous target variables.

5.1.7 LSTM

Artificial neural network is a layered design of connected neurons, enlivened by natural neural network. It is not one algorithm yet mixes of different algorithm which permits us to do complex procedure on data.

Recurrent Neural Networks (RNN) is a class of neural networks customized to manage worldly information. The neurons of RNN have a cell state/memory, and input is handled by this interior state, which is accomplished with the assistance of loops within the neural networks. There is repeating module of 'tanh' layers in

RNNs that permit them to hold data but not for too long. This is the reason we need LSTM models.

LSTM are special recurrent neural network that can capture long term dependencies. The cell state is regulated using gates which determine the amount of information that will flow through them. It has cell state c_t along with hidden states which stores information. This information can travel through cell state without any change hence, preserving long term dependencies.

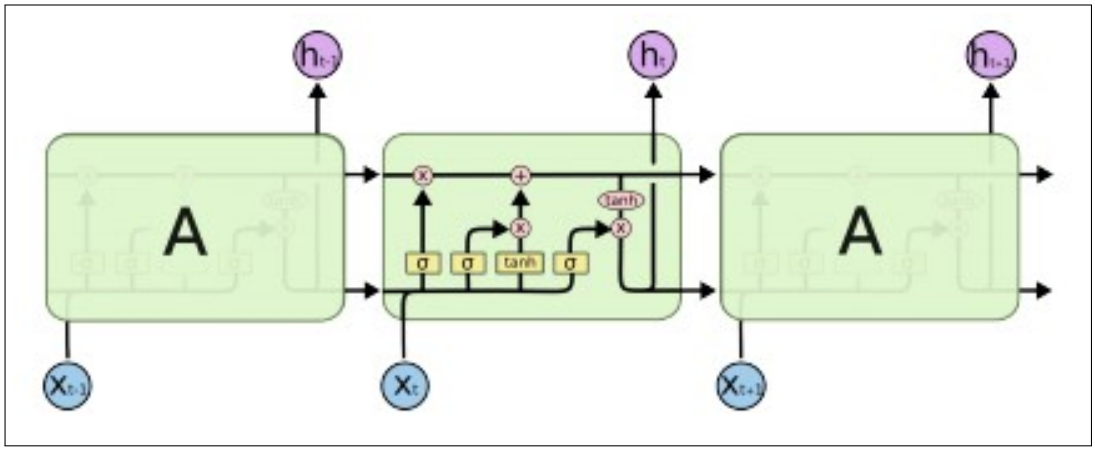

Figure 5.3: LSTM Model

RNN and LSTM give output based on current data and past data that have already passed through it. One directional LSTM does not take in account data further in sequence while predicting. Bi-directional LSTM trains two independent LSTMs in opposite directions and connect the both the hidden layers to the same output. One LSTM trains in forward direction and other in backward direction. Using the two hidden states combined we can use information from both past and future. In prediction of next word problem, Unidirectional LSTM can only see "The girl went to . . . " but in Bidirectional LSTM, forward LSTM sees "The girl went to . . . " and a backward LSTM sees ". . . and then there was sandstorm". This information provided by Backward LSTM can help to understand what next word is.

A single LSTM unit contains the following layers :

- Input Gate layer
- Forget Gate layer

- Output Gate layer
- Candidate layer
- Output layer

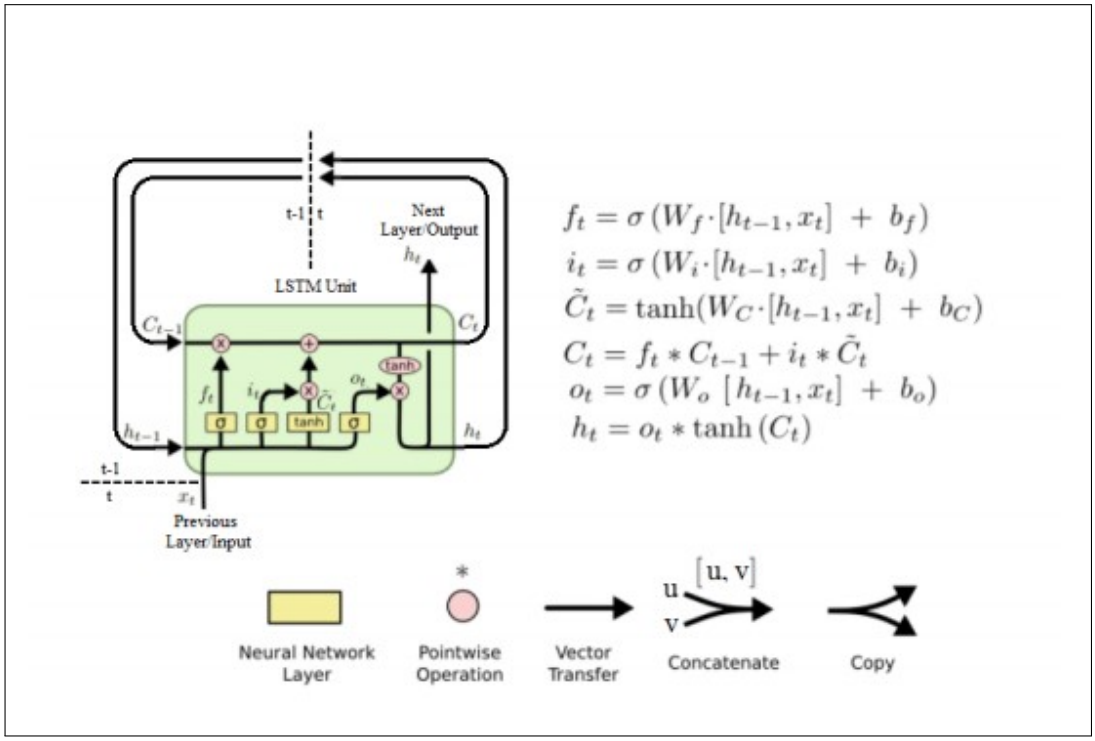

Figure 5.4: Detailed LSTM Model

All the gate layers use sigmoid as the activation function. The candidate and output layers use tanh activation function. Candidate layer takes Ht1 and Xt as input and output candidate cell state(C t) based on the given current input and previous hidden state.

$$C_t^- = tanh(W_c.[h_{t1}, x_t] + b_f)$$

Forget gate (ft) determines how much the previous cell state will be forgotten. The forget gate layer takes current input and previous hidden state as input and outputs a number between 0 to 1. The resulting number is later element-wise multiplied by previous cell state.

$$f_t = \sigma(W_f.[h_t - 1, x_t] + b_f)$$

Input layer also takes current input and previous hidden state as input and outputs a number(It) between 0 to 1. The input gate regulates the amount by which candidate cell state will affect the actual cell state.

$$i_t = \sigma(W_i.[h_t - 1, x_t] + b_i)$$

Output layer generates the candidate output by applying tanh activation function on current cell state. Output gate layer takes current input and previous hidden state as input and outputs a number between 0 to 1. The final output is obtained by multiplying the output generated by the output layer and the output gate

$$o_t = \sigma(W_o.[h_t - 1, x_t] + b_o)$$

The cell state and hidden state are modified as following : $C_t^- =$

$$h_t = o_t * tanh(C_t)$$

5.2 Software Used

We have utilized Google Colab GPU Runtime for building of complete project in which Jupyter notebook can be executed.

The Jupyter Notebook is an open source web application that you can use to create and share documents that contain live code, equations, visualizations, and text. Jupyter Notebook is maintained by the people at Project Jupyter.

Jupyter Notebooks are a spin-off project from the IPython project, which used to have an IPython Notebook project itself. The name, Jupyter, comes from the core supported programming languages that it supports: Julia, Python, and R. Jupyter ships with the IPython kernel, which allows you to write your programs in Python, but there are currently over 100 other kernels that you can also use.

Colab is a free Jupyter notebook environment that runs entirely in the cloud. Most importantly, it does not require a setup and the notebooks that you create can be simultaneously edited by your team members - just the way you edit documents in Google Docs. Colab supports many popular machine learning libraries which can be easily loaded in your notebook.

5.2.1 Features

As a programmer, you can perform the following using Google Colab.

- Write and execute code in Python
- Document your code that supports mathematical equations
- Create/Upload/Share notebooks
- Import/Save notebooks from/to Google Drive
- Import/Publish notebooks from GitHub
- Import external datasets e.g. from Kaggle
- Integrate PyTorch, TensorFlow, Keras, OpenCV
- Free Cloud service with free GPU

5.3 Hardware Specification

- RAM - 16 GB Ram
- Disk Space - 70 GB
- GPU - NVIDIA Persistence-M 16 GB Ram

5.4 Programming Langauge Used

Python

5.5 Components

- Visualization of DataSet
- Logisitc Regression Module
- SVM Module
- LSTM Module with Glove and Word2Vec Embeddings

5.6 Coding Style Format

Python uses indentation to indicate control structures, so correct indentation is required. By doing this, the need for bracketing with curly braces (i.e. and) is eliminated. On the other hand, copying and pasting Python code can lead to problems, because the indentation level of the pasted code may not be the same as the indentation level of the current line. Such reformatting can be tedious to do by hand, but some text editors and IDEs have features to do it automatically.

Chapter 6

TEST DATA SETS, RESULTS AND ANALYSIS

After applying 3 different machine learning models to our dataset, we got the result in form of ROC AUC, Accuracy, Hamming Loss and Log Loss.

Model	ROC AUC Score	Hamming Loss	Log Loss
Logistic Regression	0.74	0.03	1.01
SVM Binary Relevance	0.68	0.03	1.62
SVM Classifier Chains	0.70	0.03	1.51
LSTM (Word2Vec Embedding)	0.96	0.02	0.28
LSTM (Glove Embedding)	0.96	0.02	0.29

6.1 Analysis

After analysis results, we can say that LSTM with Glove embedding performs the best because it has highest ROC AUC score and lowest Hamming loss and one of the lowest Log loss among all models which means that there is very less multilabel are accurately measured. LSTM with Word2Vec embedding has also performed comparable to LSTM with glove embedding. We also observe that Classifier SVM performs better than Binary Relevance SVM which was expected. Both hamming loss and log loss in all our models are lower than the algorithms presented in [20]. It was expected for deep learning model LSTM to have best result over all the algorithms.

Chapter 7

TESTING

7.1 Unit Testing

Unit Testing is a software testing approach by which individual units of source code, sets of modules together with concerned control data, usage procedures, and operating procedures are tested to determine if they are fit for usage. A unit is considered as the smallest testable part of an application. It could be either an entire module or more commonly an individual function or procedure. A unit is often an entire interface such as a class in the case of object-oriented programming, but at the same time could be an individual method. The programmer's fragment source code into short fragments occasionally by white box testers during the development phase. Ideally, each test case must be independent of the other. Substitutes such as method stubs, fakes, test harnesses can be used to help to test a module in quarantine. Unit tests are designed and run by software developers to ensure that code meets its design specifications and performs the operation as intended. We performed unit testing on the all the utility functions independently.

7.2 Integration Testing

Integration testing is the software testing process that comes after unit testing. In this testing, individual components are tested in modules. The purpose of integration testing is two find out any defect that can occur while integrating different modules. We performed integration testing on combination of units being used in the preprocessing step and training step to read and convert the audio into spectograms

7.3 System Tests

System testing is testing conducted on a complete integrated system to evaluate the system's compliance with its specified requirements. System testing takes, as its input, all of the integrated components that have passed integration testing. System testing seeks to detect defects both within the "inter-assemblages" and also within the system as a whole. The actual result is the behavior produced or observed when a component or system is tested.

Test Case	Test Case Objective	Expected Result	Actual Result	Status
TC-1	To test that the project is running in the appropriate environment.	All the required binaries should be installed. Module not found error should not be raised.	All required modules were installed and no errors were raised.	Pass
TC-2	To test whether the input audio is in correct format	Audio files should be in .wav format and invalid file format exception should not be raised.	The audio files were in .wav format and no error were raised.	Pass
TC-3	To test whether the audio files are correctly separated according to class - Depressed and Non-Depressed .	Two folders should be created automatically - Depressed and Nondepressed each of which will contain spectrogram images of respective samples.	Two folders were created each having their respective spectrogram images verified using Participant ID.	Pass
TC-4	To test the entire flow for audio processing, feature extraction, conversion to spectrogram, training and display result generation.	The CNN model should be able to read the images from the training set and be able to start training and print the output such as accuracy and TP, TN, FN, FP.	The CNN model was getting trained and we can see the output being printed accordingly.	Pass

Figure 7.1: Test Case Table

Chapter 8

CONFIGURATION MANAGEMENT PLAN

8.1 Roles and Responsibilities

The following roles and responsibilities pertain to the Plan for building our project.

8.1.1 Configuration Management

- Decided configuration standards and templates
- Designed whole system that we had to build
- Decided flow of the work
- Managed deadlines for the modules and their integration.

8.1.2 Developers

The developers were responsible for:

- Deciding all the use cases that has to be included.
- Deciding best suited softwares and tools for the project.
- Learning use of new softwares or tools needed in project
- Building first version of project and maintaining configuration of code.
- Testing every module built and then whole system on integration of modules.
- Fixing all the bugs and maintaining code after alpha release of the engine.

8.1.3 Project Management

- Designating work to the members in the group to develop.
- Keeping track of deadlines, if any of it is delayed then managing shift of man-power.
- Checking the project being build with the use cases that had to be covered.
- Once the build was complete, then assigning people for testing.
- After confirming the build of software, preparing paper with IEEE standards.
- Checking plagiarism and then publishing paper based on engine built..

Chapter 9

SOFTWARE QUALITY ASSURANCE PLAN

9.1 Quality Objective

- To help the customer in detecting and identifying toxic comments easily.
- The customer should get reasonably accurate results for his queries.
- The toxic comment detection system should justify its decision by labelling the comment quried for as toxic or non-toxic along with the type.

9.2 Durability

- We have chosen Logistic Regression , SVM , LSTM and Bi-directional LSTM along with popular glove embedding for toxic comment detection and identification to support our model.
- This project takes into consideration customer's every expectation conveniently displaying the results.
- The project uses efficient models to perform the core task of comment detection which makes the project quite accurate and efficient for the customer to use.
- The project has a very simple architectural design and flow of information and hence the maintainability of this project will be quite easier than the rest of the

toxic comment detection deployments.

9.3 Integration

- The project deals with the detection and idetifiaction of toxic comments giving a wide scope for this project to be integrated with APIs or exposed as an independent website.
- Further integration to this project could be creating an API to be able to communicate from various projects in a RESTful manner.
- Integrating a feedback mechanism could help the project to be upgraded according to the customer's need so that it satisfies their expectations.

9.4 Upgradation

- Creating a support chat to allow the user to ask any queries in the platform itself.
- Course and fine detection of manipulated objects.

Chapter 10

CONCLUSION

We compared the performance of the model based on the mean ROC AUC scores, hamming and log loss. Hamming and Log loss of classical models are more than deep learning LSTM model. We also found out that the classifier chain method performed slightly better than binary relevance in this task. LSTM model outperformed other models in this task. We can further test the performance of state of the art models like Transformers on this task. BERT (Bidirectional Encoder Representations from Transformers) caused stir in Machine Learning community by presenting state-of-the-art results in wide NLP tasks. BERT can be used for this problem in future. We can also experiment with more sophisticated models like GRUs. We can also combine machine learning models together for this problem. We can ensemble the results obtained from the various models in a majority vote fashion

Chapter 11

REFERENCES

1. J. Johnson, ”statista,” 27 Jan 2021. [Online]. Available: https://www.statista.com/statistics/273018/number-of-internet-users-worldwide/. [Accessed 15 Mar 2021].

2. B. S., ”The Role of Social Media in Mobilizing People for Riots and Revolutions,” in Social Media in Politics, Public Administration and Information Technology, vol 13. https://doi.org/10.1007/978-3-319-04666-$2_19, Springer, Cham$, 2014.

3. R. Assainar, ”The Hindu,” 04 May 2020. [Online]. Available: https://www.thehindu.com/news/cities/mumbai/kalamboli-man-abuses-police-on-fb-booked/article31496732.ece.

4. A. Bharadwaj, ”The Hindu,” 12 August 2020. [Online]. Available: https://www.thehindu.com/news/national/at-least-three-killed-in-police-firing-as-riots-break-out-over-fb-post-in-bengaluru/article32331790.ece.

5. K. Dilanian, ”NBC News,” 8 March 2021. [Online]. Available: https://www.nbcnews.com/politics/justice-department/fbi-official-told-congress-bureau-can-t-monitor-americans-social-n1259769.

6. C. C. J. B. K. R. V. Alexandra Olteanu, ”The Effect of Extremist Violence on Hateful Speech Online,” in Twelfth International AAAI Conference , 2018.

7. M. Duggan, ”Pew Research,” July 2017. [Online]. Available: https://www.pewresearch.org/internet/wp-content/uploads/sites/9/2017/07/$PI_2017.07.11_{Online-Harassment_FINAL.pdf}$.

8. J. F. T. P. A. R. A. J. K. Marilyn Walker, "A Corpus for Research on Deliberation and Debate," in Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12), Istanbul, 2012.

9. P. S. H. T. S. R. P. S. S. K. M. P. G. A. M. Binny Mathew, "Thou shalt not hate: Countering Online Hate Speech," in ICWSM 2019, 2019.

10. J. T. A. T. Y. M. Y. C. Chikashi Nobata, "Abusive Language Detection in Online User Content," in WWW '16: Proceedings of the 25th International Conference on World Wide Web, 2016.

11. S. K. V. T. M. IKONOMAKIS, "Text Classification Using Machine Learning Techniques," WSEAS TRANSACTIONS on COMPUTERS, vol. 4, no. 8, 2005.

12. M.-L. N. Huy Nguyen, "A Deep Neural Architecture for Sentence-level Sentiment Classification in Twitter Social Networking," in PACLING Conference , 2017.

13. J. W. K. R. L. X. Z. Liang-Chih Yu, "Refining Word Embeddings for Sentiment Analysis," in Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, 2017.

14. N. T. L. D. Ellery Wulczyn, "Ex Machina: Personal Attacks Seen at Scale," in WWW '17: Proceedings of the 26th International Conference on World Wide Web, 2017.

15. S. K. B. Z. R. P. Hossein Hosseini, "Deceiving Google's Perspective API Built for Detecting Toxic Comments," in Computers and Society (cs.CY); Social and Information Networks (cs.SI), 2017.

16. Y. Kim, "Convolutional Neural Networks for Sentence Classification," in Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), 2014.

17. Y. Chen, Y. Zhou, S. Zhu and H. Xu, "Detecting Offensive Language in Social Media to Protect Adolescent Online Safety," in 2012 International Conference on Privacy, Security, Risk and Trust and 2012 International Confernece on Social Computing, Amsterdam, Netherlands, 2012.

18. N. Chakrabarty, ”A Machine Learning Approach to Comment,” in INTERNATIONAL CONFERENCE ON COMPUTATIONAL INTELLIGENCE IN PATTERN RECOGNITION (CIPR 2019), 2019.

19. colah, ”Colah Blog,” 27 August 2015. [Online]. Available: https://colah.github.io/posts/2015-08-Understanding-LSTMs/. [Accessed 18 03 2021].

20. H. K. J. H. G. S. Rahul, ”Classification of Online Toxic Comments Using Machine Learning Algorithms,” in Proceedings of the International Conference on Intelligent Computing and Control Systems , 2020.

Annexure A

PLAGIARISM REPORT

Plagiarism report is attached from next pages.

Report Toxic Comment Analysis

ORIGINALITY REPORT

59%	55%	9%	13%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	www.ijert.org Internet Source	42%
2	Submitted to Army Institute of Technology Student Paper	3%
3	Submitted to Savitribai Phule Pune University Student Paper	2%
4	www.tutorialspoint.com Internet Source	2%
5	realpython.com Internet Source	2%
6	en.wikipedia.org Internet Source	1%
7	Submitted to University of Nizwa Student Paper	1%
8	Submitted to An-Najah National University Student Paper	1%
9	Submitted to Sogang University Student Paper	1%

10	coeit.kces.in Internet Source	1 %
11	link.springer.com Internet Source	<1 %
12	Submitted to Kingston University Student Paper	<1 %
13	Submitted to London School of Economics and Political Science Student Paper	<1 %
14	www.slideshare.net Internet Source	<1 %
15	dokumen.pub Internet Source	<1 %
16	"Deep Learning-Based Approaches for Sentiment Analysis", Springer Science and Business Media LLC, 2020 Publication	<1 %
17	Submitted to Arab Open University Student Paper	<1 %
18	towardsdatascience.com Internet Source	<1 %
19	Submitted to Asia Pacific University College of Technology and Innovation (UCTI) Student Paper	<1 %
	happydaycards.org	

20	Internet Source	<1%
21	Submitted to University of Pune Student Paper	<1%
22	"Computational Intelligence in Pattern Recognition", Springer Science and Business Media LLC, 2020 Publication	<1%
23	www.safaribooksonline.com Internet Source	<1%
24	"Natural Language Processing and Chinese Computing", Springer Science and Business Media LLC, 2018 Publication	<1%
25	Submitted to South Bank University Student Paper	<1%
26	Submitted to University College London Student Paper	<1%
27	Young-Seob Jeong, Kyo-Joong Oh, Chung-Ki Cho, Ho-Jin Choi. "Pseudo-random number generation using LSTMs", The Journal of Supercomputing, 2020 Publication	<1%
28	bmcbioinformatics.biomedcentral.com Internet Source	<1%

29	S.V. Kogilavani, S. Malliga, K.R. Jaiabinaya, M. Malini, M. Manisha Kokila. "Characterization and mechanical properties of offensive language taxonomy and detection techniques", Materials Today: Proceedings, 2021 Publication	<1 %
30	arxiv.org Internet Source	<1 %
31	Saeed - Ul Hassan, Hajra Waheed, Naif R. Aljohani, Mohsen Ali, Sebastián Ventura, Francisco Herrera. "Virtual learning environment to predict withdrawal by leveraging deep learning", International Journal of Intelligent Systems, 2019 Publication	<1 %
32	docshare01.docshare.tips Internet Source	<1 %
33	export.arxiv.org Internet Source	<1 %
34	open.library.ubc.ca Internet Source	<1 %
35	www.geeksforgeeks.org Internet Source	<1 %
36	Rahul, Harsh Kajla, Jatin Hooda, Gajanand Saini. "Classification of Online Toxic	<1 %

Comments Using Machine Learning Algorithms", 2020 4th International Conference on Intelligent Computing and Control Systems (ICICCS), 2020
Publication

37	Lecture Notes in Computer Science, 2015. Publication	<1%
38	Odysseas Papapetrou, Sebastian Michel, Matthias Bender, Gerhard Weikum. "Chapter 21 On the Usage of Global Document Occurrences in Peer-to-Peer Information Systems", Springer Science and Business Media LLC, 2005 Publication	<1%

Exclude quotes	On	Exclude matches	Off
Exclude bibliography	On		

Annexure VII: Report Documentation

Report Documentation

Report Code: CS-BE-Project 2020 -2021	**Report Number: 2**

Report Title: Toxic Comment Detection and Classification

Address (Details): Army Institute of Technology, Alandi Road, Dighi Hills, Pune PIN CODE - 160101

Author 1 [with Address, E- mail]:	**Author 2 [with Address, E- mail]:**	**Author 1 [with Address, E- mail]:**	**Author 1 [with Address, E-mail]:**
Address **Army Institute of Technology, Pune** **E-mail :** **ajaykumar_17061@aitpune.edu.in** **Roll:3406**	**Address** **Army Institute of Technology, Pune** **E-mail :** **kumarshivam_17236@aitpune.edu.in** **Roll:3449**	**Address** **Army Institute of Technology, Pune** **E-mail :** **nareshkumar_17232@aitpune.edu.in** **Roll:3455**	**Address** **Army Institute of Technology, Pune** **E-mail :** **rahulmalhan_17183@aitpune.edu.in** **Roll:7417**

Year: 2020 – 2021
Branch: Computer Engineering

***Key Words*:** Toxic Comments, Natural Language Processing, Machine Learning, Deep Learning, Text Classification, Multilabel Classification

Type of Report: FINAL	Report Checked By: Prof Sharayu Lokhande	Report Checked Date: 05/06/2021	**Guides Complete Name:** Prof Sharayu Lokhande	Total Copies 1

Abstract:
Toxic comments refers to hatred online comments classified as disrespectful or abusive towards individual or community. With a boom of internet, lot of users are brought to online social discussion platforms. These platforms are created to exchange ideas, learning new things and have meaningful conversations. But due to toxic comments many users are not able to put their points in online discussions. This degrades quality of discussion. In this paper we will check the toxicity of comment. And if the comment is toxic then classify the comments into different categories to examine the type of toxicity. We will utilize different machine learning and deep learning algorithms on our dataset and select the best algorithms based on our evaluation methodology. Moving forward we seek to attain high performance through our machine learning and deep learning models which will help in limiting the toxicity present on various discussion sites.

NOTE –
This table should not go beyond this page.
Scale down the Abstract if it does not fit in one page.
Take guide's Signature in the **"Report Checked By**:" Cell and Date of Signature in the "**Report Checked Date:**" Cell.
This page is the last page of the projects report and is NOT to be included in the **"Page Count"**