PREMIER UNIVERSITY, CHATTOGRAM

Department of Computer Science & Engineering

MLL(CSE 454) Final Project Report

On

TOXIC COMMENTS CLASSIFICATION

SUBMITTED BY

Name: Mahmudur Rashid Mehraj

ID: 1903610201794

Name: Abidur Rahman Bhuiyan

ID: 1903610201814

Name: Md Hafizul Islam

ID: 1903610201822

SUBMITTED TO

JANNATHUL MAOWA HASI

Lecturer

Department of Computer Science & Engineering

Premier University, Chattogram

16 September, 2023

Table of Contents

1 Introduction **1**

1.1 Introduction 1

1.2 Background 1

1.3 Motivation 2

1.4 Objective 2

2 Literature Review **3**

2.1 Previous Works 3

3 Methodology **5**

3.1 Data Description 5

3.2 Preprocessing 6

3.3 Feature Extraction 7

3.4 Classification 8

4 Results and Discussion **11**

4.1 Experiment and Results 11

4.1.1 Performance Measure 11

4.1.2 Results 12

4.2 Conclusion and future work: 13

Bibliography **14**

Abstract

This report introduces various machine learning approaches applied to the task of classifying toxicity in online comments. We study the impact of Support Vector Machines (SVM), Long Short-Term Memory Networks (LSTM), Convolutional Neural Networks (CNN), and Multilayer Perceptron (MLP) methods, in combination with word and character-level embeddings, on identifying toxicity in text. We evaluated our approaches on Wikipedia comments from the Kaggle Toxic Comments Classification Challenge dataset. Our word-level assessment found that our forward LSTM model achieved the highest performance on both binary classification (toxic vs non-toxic) and multi-label classification (classifying specific kinds of toxicity) tasks. Regarding character-level classification, our best results occurred when using a CNN model. Overall, however, our word-level models significantly outperformed our character-level models. Moving forward, we seek to attain higher performance through applying richer word/character representations and using more complex deep learning models. To protect users from being exposed to offensive language on online forums or social media sites, companies have started flagging comments and blocking users who are found guilty of using unpleasant language. Several Machine Learning models have been developed and deployed to filter out the unruly language and protect internet users from becoming victims of online harassment and cyberbullying.

Keywords: natural language processing, sentence classification, multi-label classification, deep learning, convolutional neuralnetwork.

CHAPTER 1 INTRODUCTION

1.1 Introduction

Online forums and social media platforms have provided individuals with the means to put forward their thoughts and freely express their opinion on various issues and incidents. In some cases, these online comments contain explicit language which may hurt the readers. Comments containing explicit language can be classified into myriad categories such as Toxic, Severe Toxic, Obscene, Threat, Insult, and Identity Hate. The threat of abuse and harassment means that many people stop expressing themselves and give up on seeking different opinions.

1.2 Background

Sentiment classification regarding toxicity has been intensively researched in the past few years, largely in the context of social media data where researchers have applied various machine learning systems to try and tackle the problem of toxicity as well as the related, more well-known, task of sentiment analysis. Comment abuse classification research initially began with Yin et al's application of combining TF-IDF with sentiment/contextual features. They compared the performance of this model with a simple TF-IDF model and reported a 6% increase in Fl score of the classifier on chat style datasets (Kongregate, MySpace). We discuss further related works specific to our approaches below.

1.3 Motivation

Recently cyber-bullying and online harassment have become two of the most serious issues in many public online communities.Conversational toxicity is an issue that can lead people both to stop genuinely expressing themselves and to stop seeking others' opinions out of fear of abuse/harassment.Unfortunately, cyber-bulling and harassment havebecome serious issues affecting a wide range of users causingdifferent psychological problems like depression and evensuicidality.So, we should work about about it.

1.4 Objective

In this chapter, we have discussed background and motivation about the toxic comments classification. The process for this project was as follows:
Analyze the problem and propose a useful solution. Explore the dataset to get a better picture of how the labels are distributed, how they correlate with each other, and what defines toxic or clean comments. Develop an objective that fits a practical use case and addresses the major class imbalance. Create a baseline score with a simple logistic regression classifier. Explore the effectiveness of multiple machine learning algorithms. Select the best model based on a balance of performance and efficiency. Refine the preprocessing strategies to optimize model performance. Tune model parameters to maximize performance. Build a the final model with the best performing algorithm and parameters and test it on a holdout subset of the data.

CHAPTER 2 LITERATURE REVIEW

2.1 Previous Works

Text classification is one of the most important problemsin natural language processing. It has been widely studiedin multiple research projects that achieved good classifica-tionaccuracy with the aid of different machine learning methods.Recently, deep learning models have been increasingly used intext classification due to their high performance and minimalneed for features engineering.

Convolutional neural networks (CNNs) have been widelyapplied for classification problems in different fields includingtext classification.To justify this decision we choose to compare CNNs against the traditional bag-of-words approach for text analysis combined with a selection of algorithms proven to be very effective in text classification. The reported results provide enough evidence that CNN enhance toxic comment classification reinforcing research interest towards this direction.

Recently Georgakopoulos et al. tackled the problem of toxiccomments classification using the same Wikipedia dataset weuse [2]. However, they proposed a CNN-based model thatonly predicts whether or not a comment is toxic. They do notaddress the problem of finding the types of toxicity present inthe comment. Their solution also avoids the problem of dataclasses imbalance by using a balanced subset of the data forbuilding the model. On the other hand, our proposed solutionpredicts the different types of toxicity in the input commentand we use data augmentation to overcome the imbalancedclasses distribution in the training data.

CHAPTER 3

METHODOLOGY

3.1 Data Description

At the end of 2017 the Civil Comments platform shut down and chose make their 2m public comments from their platform available in a lasting open archive so that researchers could understand and improve civility in online conversations for years to come. Jigsaw sponsored this effort and extended annotation of this data by human raters for various toxic conversational attributes.

In the data supplied for this competition, the text of the individual comment is found in the comment_text column. Each comment in Train has a toxicity label (target), and models should predict the target toxicity for the Test data. This attribute (and all others) are fractional values which represent the fraction of human raters who believed the attribute applied to the given comment. For evaluation, test set examples with target >= 0.5 will be considered to be in the positive class (toxic).

The data also has several additional toxicity subtype attributes. Models do not need to predict these attributes for the competition, they are included as an additional avenue for research. Subtype attributes are: severe_toxicity,obscene,threat,insult,identity_attack, sexual_explicit.

3.2 Preprocessing

Data preprocessing is the process of transforming raw data into a useful, understandable format. Real-world or raw data usually has inconsistent formatting, human errors, and can also be incomplete. Data preprocessing resolves such issues and makes datasets more complete and efficient to perform data analysis.

Data cleaning
Data cleaning or cleansing is the process of cleaning datasets by accounting for missing values, removing outliers, correcting inconsistent data points, and smoothing noisy data. In essence, the motive behind data cleaning is to offer complete and accurate samples for machine learning models.The techniques used in data cleaning are specific to the data scientist's preferences and the problem they're trying to solve. Here's a quick look at the issues that are solved during data cleaning and the techniques involved.

Missing values
The problem of missing data values is quite common. It may happen during data collection or due to some specific data validation rule. In such cases, you need to collect additional data samples or look for additional datasets.The issue of missing values can also arise when you concatenate two or more datasets to form a bigger dataset. If not all fields are present in both datasets, it's better to delete such fields before merging.

Noisy data
A large amount of meaningless data is called noise. More precisely, it's the random variance in a measured variable or data having incorrect attribute values. Noise includes duplicate or semi-duplicates of data points, data segments of no value for a specific research process, or unwanted information fields.For example, if you need to predict whether a person can drive, information about their hair color, height, or weight will be irrelevant.

Numerosity reduction
Numerosity reduction is the process of replacing the original data with a smaller form of data representation. There are two ways to perform this: parametric and non-parametric methods.Parametric methods use models for data representation. Log-linear and regression methods are used to create such models. In contrast, non-parametric methods store reduced data representations using clustering, histograms, data cube aggregation, and data sampling.

3.3 Feature Extraction

Feature extraction for machine learning and deep learning. Feature extraction refers to the process of transforming raw data into numerical features that can be processed while preserving the information in the original data set. It yields better results than applying machine learning directly to the raw data.

TF-IDF

TF-IDF which stands for Term Frequency – Inverse Document Frequency. It is one of the most important techniques used for information retrieval to represent how important a specific word or phrase is to a given document. Let's take an example, we have a string or Bag of Words (BOW) and we have to extract information from it, then we can use this approach.

Term Frequency (TF):
- It demonstrates the importance of the word to a doc with an intuition that the more the term in doc means higher the importance is.
- TF = (number of times the term appears in the document) / (total number of terms in the document)

Inverse Document Frequency (IDF):
- Shows how a term is actually relevant. It is not necessary that term frequently in some docs could be relevant such as stopwords (the that, of, etc). Stopwords do not reveal the context and thus these should be avoided. IDF works in such a way that it ignores them.
- It penalizes the word appearing frequently across docs
- The IDF score is higher for the relevant term while lower weight to the stopword
- It considers natural logarithmic function aka log e
-IDF = log * ((number of the documents in the corpus) / (number of documents in the corpus contain the term))

The tf-idf value increases in proportion to the number of times a word appears in the document but is often offset by the frequency of the word in the corpus, which helps to adjust with respect to the fact that some words appear more frequently in general. TF-IDF use two statistical methods, first is Term Frequency and the other is Inverse Document Frequency. Term frequency refers to the total number of times a given term t appears in the document doc against (per) the total number of all words in the document and The inverse document frequency measure of how much information the word provides. It measures the weight of a given word in the entire document. IDF show how common or rare a given word is across all documents. TF-IDF can be computed as tf * idf

Tf*Idf do not convert directly raw data into useful features. Firstly, it converts raw strings or dataset into vectors and each word has its own vector. Then we'll use a particular technique for retrieving the feature like Cosine Similarity which works on vectors, etc. As we know, we can't directly pass the string to our model. So, tf*idf provides numeric values of the entire document for us.

3.4 Classification

a. Logistic Regression: This type of statistical model (also known as logit model) is often used for classification and predictive analytics. Logistic regression estimates the probability of an event occurring, such as voted or didn't vote, based on a given dataset of independent variables. Since the outcome is a probability, the dependent variable is bounded between 0 and 1. In logistic regression, a logit transformation is applied on the odds—that is, the probability of success divided by the probability of failure. This is also commonly known as the log odds, or the natural logarithm of odds, and this logistic function is represented by the following formulas:

Logit(pi) = 1/(1+ exp(-pi))

ln(pi/(1-pi)) = Beta_0 + Beta_1*X_1 + . . . + B_k*K_k

In this logistic regression equation, logit(pi) is the dependent or response variable and x is the independent variable. The beta parameter, or coefficient, in this model is commonly estimated via maximum likelihood estimation (MLE). This method tests different values of beta through multiple iterations to optimize for the best fit of log odds. All of these iterations produce the log likelihood function, and logistic regression seeks to maximize this function to find the best parameter estimate. Once the optimal coefficient (or coefficients if there is more than one independent variable) is found, the conditional probabilities for each observation can be calculated, logged, and summed together to yield a predicted probability. For binary classification, a probability less than .5 will predict 0 while a probability greater than 0 will predict 1. After the model has been computed, it's best practice to evaluate the how well the model predicts the dependent variable, which is called goodness of fit. The Hosmer–Lemeshow test is a popular method to assess model fit.

b. Decision Tree: Decision Tree is a Supervised learning technique that can be used for both classification and Regression problems, but mostly it is preferred for solving

Classification problems. It is a tree-structured classifier, where internal nodes represent the features of a dataset, branches represent the decision rules and each leaf node represents the outcome.
In a Decision tree, there are two nodes, which are the Decision Node and Leaf Node. Decision nodes are used to make any decision and have multiple branches, whereas Leaf nodes are the output of those decisions and do not contain any further branches. The decisions or the test are performed on the basis of features of the given dataset. It is a graphical representation for getting all the possible solutions to a problem/decision based on given conditions. It is called a decision tree because, similar to a tree, it starts with the root node, which expands on further branches and constructs a tree-like structure. In order to build a tree, we use the CART algorithm, which stands for Classification and Regression Tree algorithm. A decision tree simply asks a question, and based on the answer (Yes/No), it further split the tree into subtrees.

c. Naive Bayes: A Naive Bayes classifier is a probabilistic machine learning model that's used for classification task. The crux of the classifier is based on the Bayes theorem. The formula for Bayes' theorem is given as:
$P(A|B) = P(B|A)P(A)/P(B)$
Using Bayes theorem, we can find the probability of A happening, given that B has occurred. Here, B is the evidence and A is the hypothesis. The assumption made here is that the predictors/features are independent. That is presence of one particular feature does not affect the other. Hence it is called naive.
Naive Bayes algorithms are mostly used in sentiment analysis, spam filtering, recommendation systems etc. They are fast and easy to implement but their biggest disadvantage is that the requirement of predictors to be independent. In most of the real life cases, the predictors are dependent, this hinders the performance of the classifier.

d. K-Nearest Neighbor: K-Nearest Neighbour is one of the simplest Machine Learning algorithms based on Supervised Learning technique. K-NN algorithm assumes the similarity between the new case/data and available cases and put the new case into the category that is most similar to the available categories. K-NN algorithm stores all the available data and classifies a new data point based on the similarity. This means when new data appears then it can be easily classified into a well suite category by using K- NN algorithm. K-NN algorithm can be used for Regression as well as for Classification but mostly it is used for the Classification problems. K-NN is a non-parametric algorithm, which means it does not make any assumption on underlying data. It is also called a lazy learner algorithm because it does not learn from the training set immediately instead it stores the dataset and at the time of classification, it performs an action on the dataset.
KNN algorithm at the training phase just stores the dataset and when it gets new data,

then it classifies that data into a category that is much similar to the new data. Example: Suppose, we have an image of a creature that looks similar to cat and dog, but we want to know either it is a cat or dog. So for this identification, we can use the KNN algorithm, as it works on a similarity measure. Our KNN model will find the similar features of the new data set to the cats and dogs images and based on the most similar features it will put it in either cat or dog category.

CHAPTER 4
RESULTS AND DISCUSSION

4.1 Experiment and Results

4.1.1 Performance Measure

a. Accuracy: The accuracy metric is one of the simplest Classification metrics to implement, and it can be determined as the number of correct predictions to the total number of predictions.
It can be formulated as:
Accuracy = TP+TN/TP+FP+FN+TN
To implement an accuracy metric, we can compare ground truth and predicted values in a loop, or we can also use the scikit-learn module for this. We can use accuracy_score function of sklearn.metrics to compute accuracy of our classification model.

b. Precision: Precision, used in document retrievals, may be defined as the number of correct documents returned by our ML model. We can easily calculate it by confusion matrix with the help of following formula
Precision = TP/TP+FP
The precision metric is used to overcome the limitation of Accuracy. The precision determines the proportion of positive prediction that was actually correct. It can be calculated as the True Positive or predictions that are actually true to the total positive predictions (True Positive and False Positive).

c. Recall: Recall may be defined as the number of positives returned by our ML model. We can easily calculate it by confusion matrix with the help of following formula
Recall = TP/TP+FN
It is also similar to the Precision metric; however, it aims to calculate the proportion of actual positive that was identified incorrectly. It can be calculated as True Positive or predictions that are actually true to the total number of positives, either correctly predicted as positive or incorrectly predicted as negative (true Positive and false negative).

d. F1-Score: This score will give us the harmonic mean of precision and recall. Mathematically, F1 score is the weighted average of the precision and recall. The best value of F1 would be 1 and worst would be 0. We can calculate F1 score with the help of following formula
F1 Score = 2 * Precision * Recall / (Precision + Recall)
F1 score is having equal relative contribution of precision and recall. We can use classification_report function of sklearn.metrics to get the classification report of our classification model. F-score or F1 Score is a metric to evaluate a binary classification model on the basis of predictions that are made for the positive class. It is calculated with the help of Precision and Recall. It is a type of single score that represents both Precision and Recall. So, the F1 Score can be calculated as the harmonic mean of both precision and Recall, assigning equal weight to each of them.

4.1.2 Results

After using tf-idf vectorizer, we split the dataset to 70% training data and 30% test data to test the accuracy of the classifiers.

The table showing the results of Algorithm we used in [3.1]

Method used	Accuracy	Precision	Recall	F1-score
Naive Bayes	0.86	0.89	0.90	0.90
Logistic Regression	0.89	0.93	0.92	0.91
Decision Tree	0.92	0.92	0.92	0.92
KNN	0.95	0.95	0.95	0.95

Table 4.1. Result of different Algorithm.

4.2 Conclusion and future work:

Our report has shown that harmful or toxic comments in the social media space have many negative impacts to society. The ability to readily and accu-rately identify comments as toxic could provide many benefits while mitigating the harm. Also, our research has shown the capability of readily available algo-rithms to be employed in such a way to address this challenge. In our specific study, it was demonstrated that an LSTM solution provides substantial improve-ment in classification versus a baseline Na¨ıve Bayes based solution. To recall, the True Positive Rate of LSTM was almost 20% higher than Naive Bayes method. Additionally, the followings are some suggested studies to be considered as fu-ture work in this area: a) We suggest a plan to improve the NLP classifiers: first by using other algorithms which such as Support Vector Clustering (SVC) and Convolutional Neural Networks (CNN); secondly, extend the classifiers to the overall goal of Kaggle competition which is multi-label classifiers. in the current study, the problem simplified into two classes but it worth to pursue a main goal which is 7 classes of comments. b) We also suggest using SVM for text processing and text classification. It requires a grid search for hyper-parameter tuning to get the best results.

BIBLIOGRAPHY

[1] *Dataset collected form kaggle.* Dataset link: https://www.kaggle.com/competitions/jigsaw-toxic-comment-classification-challenge/data (accessed on: 08 Sepetember 2022).

[2] *Previous work i.* Access link: https://scholar.smu.edu/datasciencereview/vol3/iss1/13/ (accessed on: 03 Sepetember 2022).

[3] *Previous work ii.* Access link: https://ieeexplore.ieee.org/abstract/document/9120939 (accessed on: 04 Sepetember 2022).