Cyberbullying Detection using Bengali comments from social media platform

by

Parom Guha Neogi
20101562
Md. Jisan Mashrafi
21301058
Rakibul Hasan
20301300
Md Shadman Shakib
20301127
IFTEKHAR AHMED
19201097

A project submitted to the Department of Computer Science and Engineering
School of Data and Sciences
in partial fulfillment of the requirements for the degree of
B.Sc. in Computer Science

Department of Computer Science and Engineering
Brac University
December 2023

Abstract

Majority of people nowadays use social media on a regular basis and it is clear that it has infiltrated our lives. Additionally, many people rely on social media to earn a living. Cyberbullying and vulgar language on social networking sites have become serious issues, particularly among Bengali language users. Natural language processing algorithms have a difficult time locating and identifying such problematic information due to the complexity of Bangla text data. This study provides a machine learning-based data-driven security solution for detecting and categorizing cyberbullying and toxic language for Bengali language users. We looked into and evaluated the effectiveness of several natural language processing (NLP) methods for spotting abusive language. We also provide an alternative strategy for tackling the Bangla-specific problems with current approaches. This research may enable networking sites and other online platform providers to provide web access that is secure for Bengali language users.

Keywords: Cyberbullying; Detection; Internet; Language; NLP

Table of Contents

Chapter 1

Introduction

The employees of security operations centers (SOCs) are widely utilized in present vital infrastructure vulnerability assessment methods. (2017) Feng et al. Conventional manual and subjective audits, however, fall short of the necessary qualities of dynamic CIs and attack surface assessment. For example, certain frequently used tools for determining vulnerability severity, such the CVSS1 calculator, need input from users and depend on qualitative evaluations of vulnerability features, such as exploitability, scope, and repercussions. (2011) Joh and Malaiya. However, there is a wealth of textual information about cybersecurity, including in vendor declarations, blogs, whitepapers, and hacker forums. Traditional threat analysis of data from textual sources uses the time-consuming and unproductive human labor technique. The security analysts are therefore unable to properly utilize the validated cybersecurity information to react to cyber threats in a timely and correct manner.

Chapter 2

Literature review

2.1 Related Work

Cyberbullying is a serious problem in the modern world, especially on social networking sites like Facebook. The victim of electronic intimidation or harassment may experience negative effects on their mental and emotional health. In an effort to put an end to the practice, researchers have looked into the use of natural language processing (NLP) techniques to automatically detect instances of cyberbullying on social networking sites like Facebook. This literature review looks at recent studies on Facebook's for detecting cyberbullying. It is more important to recognize cyberbullying as social networking platforms continue to grow in popularity. Many studies have used machine learning techniques to solve this issue. In one of these studies, supervised machine learning was used to ascertain the sentiment and meaning of sentences. Unfortunately, the accuracy rating of this algorithm was only 61.9%, indicating that it may not be trustworthy for detecting cyberbullying. [1]. Cyberbullying on social media is an increasing problem, making its identification and prevention essential. To detect cyberbullying on Twitter-based networks, present a supervised machine learning method[2]. Users' actions and tweet content are two factors that the authors of the study choose from Twitter and put into an identification algorithm. One of the challenges in developing such a model is the requirement for a robust and representative dataset. A sufficient number of tweets must be included in the dataset to adequately represent both cyberbullying as well as non-cyberbullying behavior.

Dinakar et al. use a variety of machine learning methods, including logistic regression, decision trees, and assistance vector machines, to develop the detection model [3]. They experiment with several feature sets and find that a combination of actions, user, and tweet material components produces the greatest results. As shown by the assessment of the detection system, which achieved the f-measure of 0.936 as well as a region within the receiver-operating curves of 0.943, it effectively detects harassment tweets with a high level of accuracy.

Zhao et al. studied the automatic detection of cyberbullying on social media: A deep learning approach[4]. Using a significant dataset and cutting-edge neural network architecture, the authors try to identify and classify various types of cyberbullying. The key difficulties tackled by this research are the variety of cyberbullying content,

the discrepancy among instances of positive and negative bullying, and the obligation to take into account the temporal and situational nature of social media posts.

Ahmed et al. used a deep learning-based technique that combines an LSTM network with a network of convolutional neural networks (CNN) to extract and contextual information from social media posts. The recommended method showed outstanding accuracy of 87 percent, outperforming several state-of-the-art techniques. The paper [5] proposes a hybrid deep learning approach for the identification of cyberbullying in social media.The authors aim to overcome the challenges of finding and categorizing cyberbullying content in large-scale social media data by harnessing the strength of deep learning and natural language processing approaches.The difficulties in developing a highly accurate cyberbullying detection system are noted as being the complexity of social media data and the need for trustworthy feature extraction. By combining a convolutional neural network (CNN) and a bidirectional long short-term memory network (BiLSTM), the proposed hybrid technique categorizes cyberbullying content. The accuracy rate the authors achieved on the dataset they used was 95.2%, outperforming state-of-the-art models at the time.

To improve accuracy and reduce false positives, Iwendi et al.[6] propose a novel deep learning approach for recognizing cyberbullying via social media. The authors battled with a number of challenges, such as the dataset's asymmetries, the variety of comments that weren't about online harassment, and the vocabulary used in social media. The suggested method is a multi-layer convolutional neural network called the Cyberbullying Detecting Network (CDN), which combines character and word embeddings. With an accuracy of 95.73%, the CDN outperformed several other state-of-the-art algorithms when it came to of recall, accuracy, and F1-score.

2.2 Research Objective

Since there have been many previous studies that have addressed cyberbullying, our major goal in doing this study is to determine how to identify cyberbullying more effectively in the Bangla language on social media platforms. The correctness of the analysis result that we obtained from running several models on our dataset is thus our key objective. We will be able to determine the context or meaning of phrases and tell which ones are uttered for humorous purposes and which ones are actually meant to bully if we can utilize some approach to comprehend human psychology. Implementing this will also enable us to increase user awareness of their social media postings, comments, and other actions. Cyberbullying is a risky and damaging practice that can result in making someone being mentally unstable and sometimes it can even lead some victims to suicidal attempts too. Based on the aforementioned problems, the researchers ought to create a system for identifying and stopping abusive behavior on digital platforms. Because of users' unacceptable behavior that appears on several social media platforms, there is a need to create a trustworthy multi-model detection system. In this research work, a summary of groundbreaking on text-based cyberbullying detection has been gathered. The current attempt aims to gather and debate research on the application of machine learning for the detection of cyberbullying by looking through the findings from

prior studies. Focusing on reviews enables us to highlight the ongoing discussion and theoretical viewpoints on the subject while also outlining the questions we need to address. Bangladesh has a sizable population, with a disproportionately large proportion of children, the unemployed, and people who feel unsafe and in need of security. But due to a lack of named dictionaries, little research has been done on social media observing the texts written in Bangla, annotated corpora, and morphological analyzers, which from Bangladesh's perspective calls for a thorough analysis. Cyberbullying has serious consequences in today's social media age. It is a touchy subject. Because Bangla was a language with few resources, it was not given the chance to be thoroughly researched. Cyberbullying is a topic that might benefit greatly from more investigation due to the small amount of data and research on it in Bangla. Researchers have also used conventional machine learning techniques in the past, but these approaches ultimately turned out to be inefficient and inaccurate. Academics have conducted a number of research and tests to determine whether cyberbullying occurs on social media sites like Wikipedia, Facebook, Twitter, and others because it is already a well-established and well-defined type of bullying. With the help of the recommended technique, transformer provides a special way to spot bullying on many social media sites. So, deep neural network based models exceeded earlier conventional models in terms of performance in the very current era. Additionally, compared to the earlier versions, it produces better outcomes. But there is always room for improvement and since the number of Bangla speakers is relatively significant, the chance of improving techniques for better outcomes has always been a welcoming fact. Although it is challenging for machine learning, we are used to writing Bangla using the English alphabet. Once more, we fall short in this area. And if we can put into practice the kind of generator that will change words written in Bangla with English alphabets into the actual Bangla word that the speaker or writer intended to express, the capacity to detect cyberbullying on social media platforms will be drastically changed by it.

Chapter 3

Explanation of analysis, requirements and design (Methodology)

The number of instances involving cyberbullying is increasing in the contemporary era of digital technology, particularly on social media platforms. Cyberbullying encompasses the act of causing injury, threat, or mistreatment against individuals through the use of digital technology. Due to the substantial volume of text that individuals submit on a daily basis, identifying instances of cyberbullying can be challenging. Researchers have been investigating how to make use of natural language processing (NLP) to automatically identify instances of cyberbullying in these sectors.

3.1 Data Collection

In the beginning of this research, we collect a dataset of social remarks and posts from research gate.The dataset is a compilation of Bengali remarks gathered from many social media platforms. This 44,000-row labeled dataset is organized into 5 distinct columns and is a valuable tool for analyzing and assessing online content. Each comment in the dataset has been meticulously annotated, with labels such as "sexual," "troll," "religious," "threat," and "non-bully." This category makes it easy to look at a wide range of subjects related to the Bengali social media landscape. It provides information on trends related to inappropriate material, trolling, religious discourse, potential threats, and stuff that isn't really bullying.

3.2 Preprocessing

Next, we do data cleaning by eliminating elements like as emojis images, and links, keeping only the textual content. To ensure text consistency, we employ techniques such as words, eliminating unnecessary terms such as "the" and "and," and conversion to lowercase. Subsequently, we proceed with obtaining crucial data from

	comment	Category	Gender	comment react number	label
0	ওই হালার পুত এখন কি মদ খাওয়ার সময় রাতের বেলা...	Actor	Female	1.0	sexual
1	ঘরে বসে শুট করতে কেমন লেগেছে? ক্যামেরাতে কে ছি..	Singer	Male	2.0	not bully
2	অরে বাবা, এই টা কোন প্যাগল????	Actor	Female	2.0	not bully
3	ক্যাপ্টেন অফ বাংলাদেশ	Sports	Male	0.0	not bully
4	পটকা মাছ	Politician	Male	0.0	troll
5	অন্যরকম .. ভালো লাগলো .. ❤️	Singer	Male	1.0	not bully
6	সাংবাদিক ভাইদের বলছি এই সংবাদ গুলি প্রচার না ক...	Actor	Female	9.0	troll
7	মোহাম্মদ কফিল উদ্দীন মাহমুদRidwan RomelDwaipay...	Actor	Female	0.0	not bully
8	ঢাকায় এত ঘনো ঘনো আগুন লাগার মূল কারন টা এতদিনে...	Actor	Female	4.0	religious
9	হিরো আলম তুমি এগিয়ে চলো, আমরা আছি তোমার সাথে।	Social	Male	0.0	not bully

Figure 3.1: Original Dataset

the cleaned dataset. The method we use includes using NLP techniques such as bag-of-words, which analyzes the text as a set of phrases, and phrase recurrence-inverse document frequency, which quantifies the significance of individual words. This enables us to convert words into numerical values that our model understands.

```
Original:
 অবশ্যই ছোট & ছেলে মেয়েকে ইসলামের শিক্ষা বাদ্যতামুলক@আফসার ভাইকে ধন্যবাদ
Cleaned:
 অবশ্যই ছোট  ছেলে মেয়েকে ইসলামের শিক্ষা বাদ্যতামুলক আফসার ভাইকে ধন্যবাদ
 Sentiment:--  religious

Original:
 কমেন্ট পড়তে আসছিলাম, এখন আমি ৭দিন পাগল থাকব
Cleaned:
 কমেন্ট পড়তে আসছিলাম  এখন আমি ৭দিন পাগল থাকব
 Sentiment:--  not bully

Original:
 সৃষ্টির ধারনা স্রষ্টার কাছ থেকেই এসেছে যা শুধু তার সৃষ্টির জন্য প্রযোজ্য,সৃষ্টার জন্য নয়।
Cleaned:
 সৃষ্টির ধারনা স্রষ্টার কাছ থেকেই এসেছে যা শুধু তার সৃষ্টির জন্য প্রযোজ্য সৃষ্টার জন্য নয়
 Sentiment:--  religious

Original:
 ও ভাই শ্লোগান দিয়েন না, আগামীকাল হরতাল হরতাল! Gazi Mansur Ahmed Rasel
Cleaned:
 ও ভাই শ্লোগান দিয়েন না  আগামীকাল হরতাল হরতাল
 Sentiment:--  troll
```

Figure 3.2: Cleaned Dataset

We want to focus solely on the text itself. To make the text more manageable, we break it down into individual words. This process is called tokenization. Additionally, we use techniques like lemmatization and stemming to normalize the data. This means reducing words to their base form so that variations of the same word are treated as one. For example, "running" and "ran" would both be considered as "run". To further refine the text, we remove stopwords. These are common words like "the" and "and" that don't add much meaning to the overall message. By eliminating stopwords, we can concentrate on the more meaningful words. Finally, we ensure consistency by converting all the remaining text to lowercase. This way, "Cyberbullying" and "cyberbullying" are treated as the same word. By preprocessing the dataset in this way, we simplify the text and remove unnecessary elements. This allows us to focus on the core content and analyze it effectively to detect cyberbullying patterns and develop detection systems on internet platforms.

3.3 Visual Representation of the Dataset

Data visualization refers to the graphical representation of information and data. Data visualization tools assist the identification and understanding of trends, outliers, and patterns in data by employing visual elements like as charts, graphs, and maps.
The 'Gender Distribution' bar plot provides insight into whether the bullying comments were directed towards males or females. Furthermore, it is abundantly evident that women are more likely than men to become the target of offensive remarks.

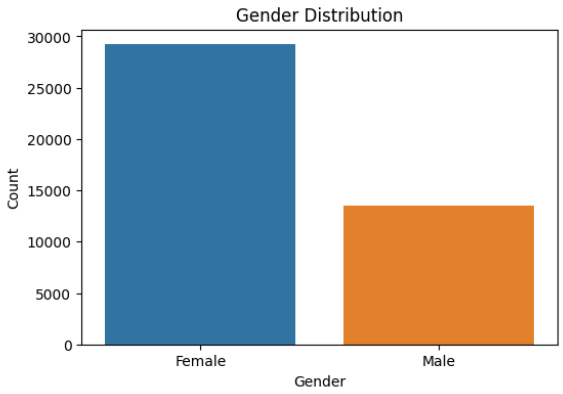


Figure 3.3: Gender Distribution

The pie chart provides an overview of the distribution of bullying comments across various categories. The Actor category is the most vulnerable to this issue, which makes up 61.3% of the chart, which is more than half. The social category takes up 21.3% of the total, making it the second biggest percentage in this chart. The categories 'Singer', 'Politician', and 'Sports' have percentages of 6.8%, 6.0%, and 4.7% respectively.

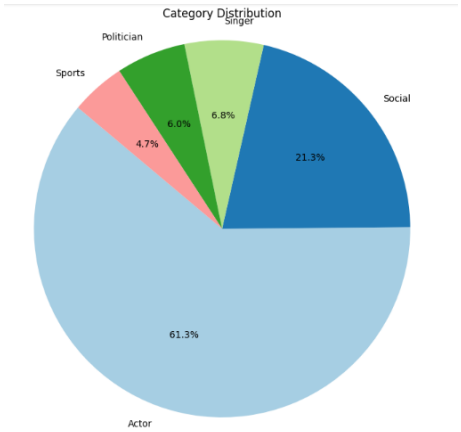


Figure 3.4: Gender Distribution

Chapter 4

Development of the project (Model Training)

4.1 Result Analysis

Result Analysis involves the building of a model with the capability of identifying instances of cyberbullying. To accomplish this, we utilize a range of machine-learning approaches. The algorithms capable of analyzing patterns in data and making predictions include Cat Boost, Multinomial Naive Bayes, logistic regression, Random Forest Classifier, XG Boost, and Support Vector Machines. In order to construct our model, we train it using the characteristics obtained from the preprocessed dataset. The objective of this stage is to develop an adaptive and efficient model capable of independently recognizing instances of cyberbullying in social media posts. Our objective is to create a model that can effectively identify and address instances of cyberbullying by utilizing various machine-learning methods and using the properties of the preprocessed dataset.

4.1.1 Cat Boost

The CatBoost library was successfully installed, and our dataset was used to train a CatBoostClassifier. For various feature sets (Unigram, Bigram, and Trigram), we have included training progress and classification reports in the output.

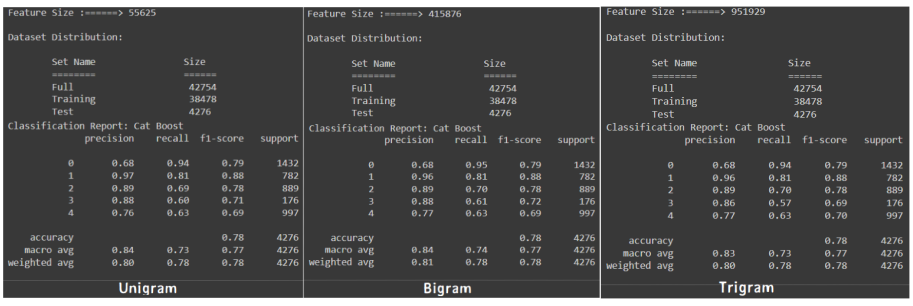


Figure 4.1: Cat Boost

Here are some key observations:
Training Time:Each feature set requires a different amount of time to train, with

Trigram taking the longest. It can take a while to train complicated models with a huge feature set.
Classification Report: The performance of the model on the test dataset is detailed in each classification report. For each class, the metrics include support, F1-score, precision, recall, macro, and weighted averages.
Accuracy: All feature sets appear to have an accuracy of about 0.78, which shows that the model is operating effectively. Accuracy by itself, though, might not give a whole view of the model's performance, particularly if the dataset is unbalanced.
Precision, Recall, and F1-score: These metrics are helpful for assessing how well the model performs for particular classes. As we can see, performance varies throughout courses, with some classes performing better in terms of precision and recall than others.
Macro and Weighted Averages: The model's performance across all classes is summarized by these averages. While the weighted average takes into account class inequalities, the macro average assigns equal weight to each class.
Feature Size: The amount of features employed in the model is indicated by the feature size for each feature set (Unigram, Bigram, and Trigram). The complexity of the feature set rises with the size of the feature.
Learning Rate: CatBoost's learning rate and iteration count are both set to 0.5 and 100, respectively. To further improve the model's performance, we can test out various hyperparameters.

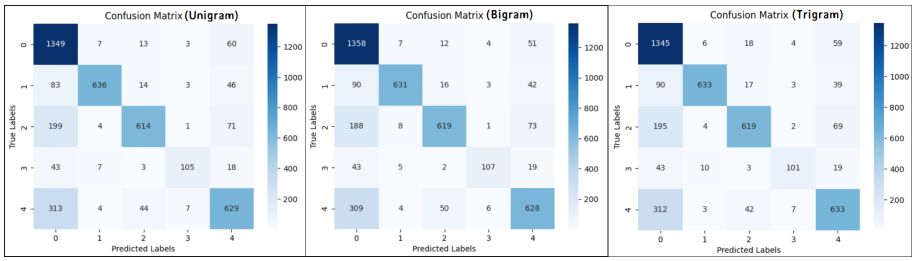


Figure 4.2: Cat Boost Confusion Matrix

Overall, based on our dataset and our evaluation of CatBoost's performance using various feature sets, it appears to be working effectively. If necessary, we may refine the model even further, experiment with other hyperparameters, or look at other options to boost its functionality.
For text categorization, CatBoost is imported and installed via pip. The processing is divided into three distinct n-gram levels (Unigram, Bigram, and Trigram). Using a CatBoostClassifier with a predetermined random seed, the classifier is trained over the course of 100 iterations. For all models, the learning rate is set at 0.5. Learnings from each iteration are shown together with the training progress. For each n-gram level, classification reports are provided, displaying precision, recall, F1-score, and total accuracy for each class.

In conclusion, CatBoost is employed to classify text on three n-gram levels, and the outcomes are shown for each level, along with measures like precision, recall, F1-score, and accuracy. The models do well in classifying text into different categories,

particularly at the Bigram and Trigram levels, where they achieve an accuracy of 78

4.1.2 Multinomial Naive Bayes

We perform text classification on three different n-gram types (unigram, bigram, and trigram) and assess the effectiveness of the 'Multinomial Naive Bayes' classifier using scikit-learn (sklearn). Machine learning problems involving text classification use the Multinomial Naive Bayes classifier. It works particularly effectively in situations when the features are discrete, such when text documents' word counts are concerned.

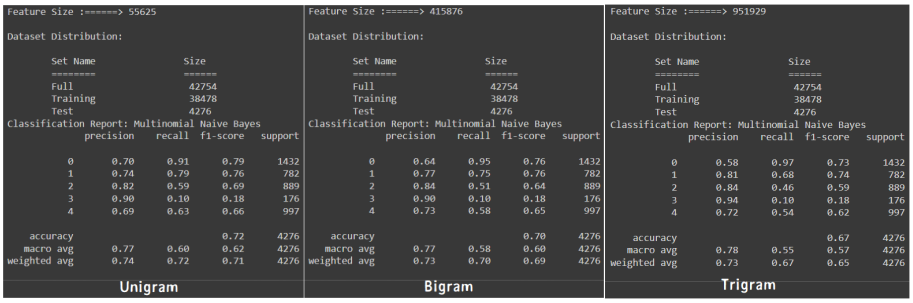


Figure 4.3: Multinomial Naive Bayes

Model Training: During training, the classifier discovers the probabilities of spotting each word or n-gram in each class. It computes the prior probability of a class in the dataset as well as the likelihood of a word or n-gram occuring in that class.Using the word or n-gram frequencies in the document, the classifier determines the likelihood that a new document will fall into each category. Afterward, it decides which class to use as the document's anticipated class by choosing the one with the highest likelihood.
Performance: Text classification tasks including spam detection, sentiment analysis, and subject categorization frequently use Multinomial Naive Bayes. It is effective and simple to construct, but the quality of the training data and the choice of features (word or n-gram representation) might have an impact on how well it performs. The code displays the model's performance for unigram, bigram, and trigram respectively.
The Multinomial Naive Bayes model's performance on the test dataset is reported in a classification report that includes statistics for each class including precision, recall, and F1-score.
Metrics like accuracy (the proportion of true positives among predicted positives), recall (the proportion of true positives among real positives), F1-score (the harmonic mean of precision and recall), and support (the number of examples in each class) are included in the classification report. The feature space and the model's capacity to recognize patterns in the text data can change based on the n-gram type utilized (unigram, bigram, or trigram), and this might affect the performance metrics.

The classification results show varying performance across n-gram models. The

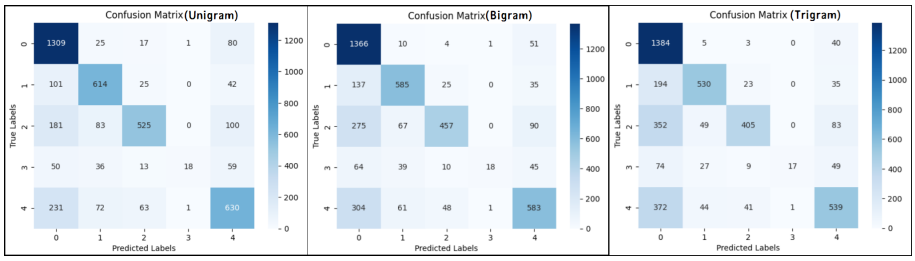
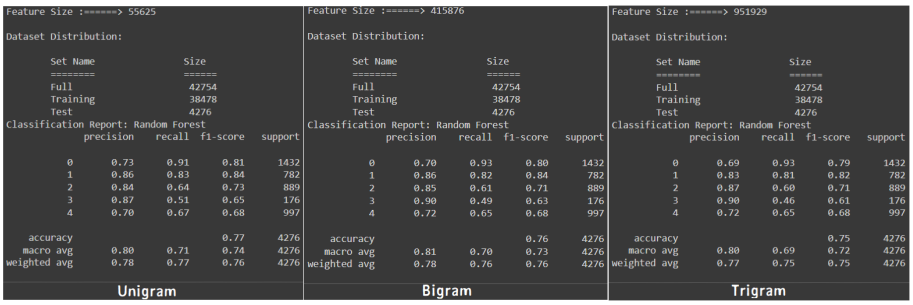


Figure 4.4: Multinomial Naive Bayes Confusion Matrix

unigram model achieves the highest accuracy (72%), capturing single-word features. The bigram model, incorporating pairs of words, slightly decreases accuracy (70%). The trigram model, considering triplets of words, further decreases accuracy (67%), indicating diminishing returns with increased context length.

4.1.3 Random Forest

Our text based dataset is trained with Random Forest Classifier with varieties of features in it such as unigram, bigram and trigram.



Feature Size :======> 55625

Dataset Distribution:

Set Name	Size
Full	42754
Training	38478
Test	4276

Classification Report: Random Forest

	precision	recall	f1-score	support
0	0.73	0.91	0.81	1432
1	0.86	0.83	0.84	782
2	0.84	0.64	0.73	889
3	0.87	0.51	0.65	176
4	0.70	0.67	0.68	997
accuracy			0.77	4276
macro avg	0.80	0.71	0.74	4276
weighted avg	0.78	0.77	0.76	4276

Unigram

Feature Size :======> 415876

Dataset Distribution:

Set Name	Size
Full	42754
Training	38478
Test	4276

Classification Report: Random Forest

	precision	recall	f1-score	support
0	0.70	0.93	0.80	1432
1	0.86	0.82	0.84	782
2	0.85	0.61	0.71	889
3	0.90	0.49	0.63	176
4	0.72	0.65	0.68	997
accuracy			0.76	4276
macro avg	0.81	0.70	0.73	4276
weighted avg	0.78	0.76	0.76	4276

Bigram

Feature Size :======> 951929

Dataset Distribution:

Set Name	Size
Full	42754
Training	38478
Test	4276

Classification Report: Random Forest

	precision	recall	f1-score	support
0	0.69	0.93	0.79	1432
1	0.83	0.81	0.82	782
2	0.87	0.60	0.71	889
3	0.90	0.46	0.61	176
4	0.72	0.65	0.68	997
accuracy			0.75	4276
macro avg	0.80	0.69	0.72	4276
weighted avg	0.77	0.75	0.75	4276

Trigram

Figure 4.5: Random Forest Classifier

The amount of decision trees in the ensemble are supplied as a parameter when creating an ensemble of decision trees for prediction purposes.

Feature Representations:This study practices the unigram, bigram and trigram featured representations. These representations convey the diverse levels of detail found in the text data. Trigrams compute triads of adjacent words, bigrams compute pairs of adjacent words and unigrams compute single words.

Classification Report: The classification report offers various indicators to assess the performance of the Random Forest Classifier on our dataset.
Metrics need support (the amount of examples in each sector), precision (the accuracy in classifying positive examples), recall(the accuracy in classifying positive examples) and F1-score(a balanced measure combining accuracy and recall). Usually, the report includes measurements for multiple classes (as this example, labeled from 0 to 4) and provides an overall accuracy rating.

Accuracy: As this categorization report, accuracy stands as a crucial measure. It shows the limit to which the classifier's prognosis on this test set was overall correct. The accuracy score is a percentage that indicates the proportion of test samples correctly identified by the model.

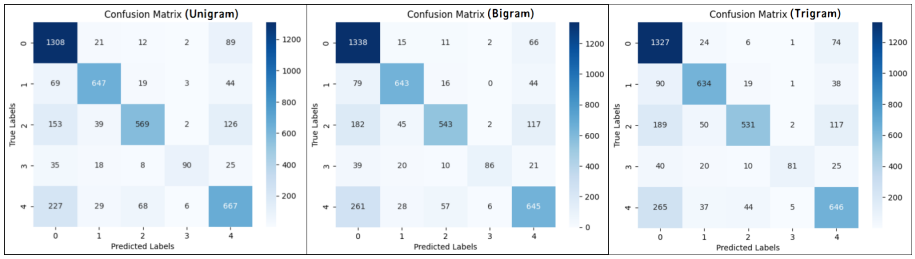


Figure 4.6: Random forest Classifier Confusion Matrix

The random forest classification results demonstrate consistent patterns across n-gram models. The unigram model achieves an accuracy of 77%, with precision and recall scores varying across classes. The bigram model maintains a similar accuracy of 76%, and the trigram model shows a slight decrease to 75%. Random forest effectively captures features from different n-gram contexts, providing robust classification performance..

4.1.4 XG Boost

Using several n-gram features (Unigram, Bigram, and Trigram), we trained an XG-Boost classifier for text classification tasks and assessed its performance on our dataset.

Feature Size :======> 55625

Dataset Distribution:

Set Name	Size
Full	42754
Training	38478
Test	4276

Classification Report: xg boost

	precision	recall	f1-score	support
0	0.70	0.95	0.81	1432
1	0.96	0.84	0.90	782
2	0.92	0.70	0.79	889
3	0.87	0.62	0.72	176
4	0.78	0.67	0.72	997
accuracy			0.80	4276
macro avg	0.85	0.76	0.79	4276
weighted avg	0.82	0.80	0.80	4276

Unigram

Feature Size :======> 415876

Dataset Distribution:

Set Name	Size
Full	42754
Training	38478
Test	4276

Classification Report: xg boost

	precision	recall	f1-score	support
0	0.71	0.95	0.81	1432
1	0.96	0.84	0.89	782
2	0.92	0.71	0.80	889
3	0.89	0.62	0.73	176
4	0.78	0.67	0.72	997
accuracy			0.80	4276
macro avg	0.85	0.76	0.79	4276
weighted avg	0.82	0.80	0.80	4276

Bigram

Feature Size :======> 951929

Dataset Distribution:

Set Name	Size
Full	42754
Training	38478
Test	4276

Classification Report: xg boost

	precision	recall	f1-score	support
0	0.70	0.95	0.81	1432
1	0.95	0.84	0.89	782
2	0.92	0.70	0.80	889
3	0.89	0.61	0.72	176
4	0.77	0.66	0.71	997
accuracy			0.80	4276
macro avg	0.85	0.75	0.79	4276
weighted avg	0.82	0.80	0.80	4276

Trigram

Figure 4.7: XG Boost

XGBoost Classifier: As help for text classification tasks, we used XGBoost classifier.
Feature extraction: Unigram, Bigram and Trigram; these three n-gram illustrations were used. Different amounts of features were extracted per n-gram illustrations.
Dataset Information: The dataset consisted of a total of 42,754 samples, which were divided into a training set comprising 38,478 samples and a test set containing 4,276 samples.
Classification Report: The main objective of the classification report was to evaluate the classifier's performance. This report encloses metrics for individual classes (0, 1,

2, 3, and 4), such as accuracy, recall, and F1-score. Additionally, aggregated scores for all classes were provided through the "macro avg" and "weighted avg" metrics. Performance Evaluation: The classifier demonstrated an overall accuracy of nearly 80% on the test set. Precision was used to evaluate the accuracy of correct predictions, while recall measured the extent of coverage for real positive cases.

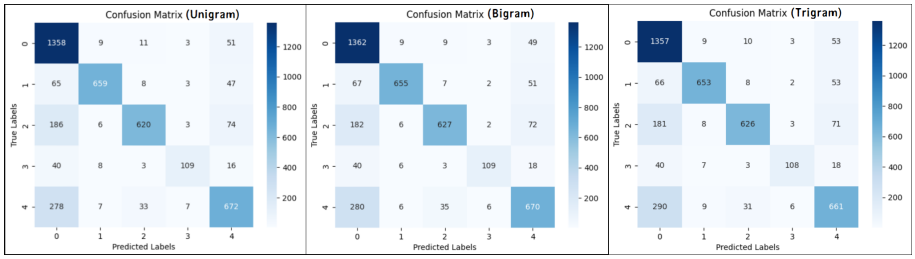


Figure 4.8: XG Boost Confusion Matrix

The XGBoost classification results exhibit consistent performance across n-gram models. The unigram, bigram, and trigram models achieve an accuracy of 80%, showcasing precision, recall, and F1-score values for each class. XGBoost effectively leverages boosting techniques to enhance classification accuracy, demonstrating its robustness across different n-gram contexts.

4.1.5 Logistic Regression

We trained a logistic regression model on various text data formats (unigram, bigram, and trigram) and assessed its effectiveness on a classification test. The code snippet starts by importing the LogisticRegression class from the scikit-learn library.

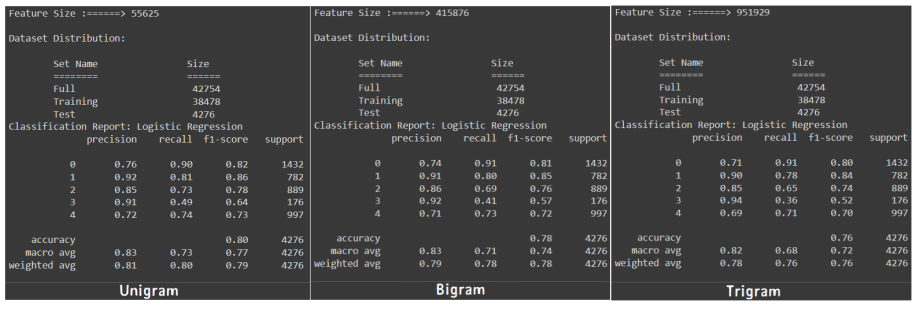


Figure 4.9: Logistic Regression

Data preprocessing: The code processes the text data, which includes unigram, bigram, and trigram data. The feature's size is decided for each type of data.
Distribution of the datasets: it shows the distribution of the dataset, incorporating the sizes of the training set, the test set, along with the complete dataset.
Report on Classification: Every data type—unigram, bigram, and trigram—has its own logistic regression model that has been trained and evaluated. Precision, recall, and F1-score for each class are included in the classification report for each case, along with accuracy in general.

The classification report is organized as follows: Precision: It evaluates the ratio of confirmed positive instances that are accurate. Recall: To refresh your memory, it is a method that counts the number of affirmative situations that were in fact positive but were appropriately F1-score: This is mainly the harmonic mean of accuracy and recall which can offer an equilibrium within the two. Evidence: There are a certain number of samples in each category.
The report provides comprehensive data on both macro and weighted averages for all classes, as well as detailed performance metrics for each class (labeled 0 to 4).
Interpretation: These metrics are used to assess the performance of the logistic regression model on trigram, bigram, and unigram text data.
How well the model assigns each of the five classes (labels 0–4) to text input is revealed by the metrics. The performance seems to be slightly affected by the type of text data used. Precision, recall, and F1-score can take on varying values depending on the class and dataset type.
The "macro avg" and "weighted avg" metrics provide a full picture of the model's effectiveness when all classes are considered. Finally, the data presented exhibits the consequences of training a logistic regression model upon different types of text input and evaluating its efficacy in a multiclass classification assignment. It assesses the model's accuracy in categorizing text samples into one of five groups and provides detailed metrics for every category and dataset type.

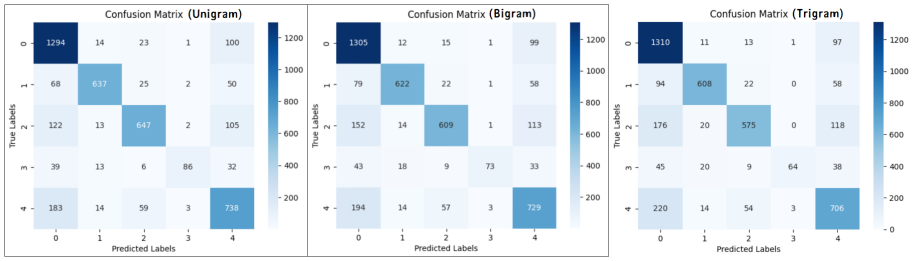


Figure 4.10: Logistic Regression Confusion Matrix

The logistic regression results consistently show improved accuracy across n-gram models. The unigram model achieves the highest accuracy at 80%, followed by the bigram model with 78%, and the trigram model with 76%. Logistic regression effectively utilizes contextual information, resulting in enhanced classification performance compared to Naive Bayes.

4.1.6 Support Vector Machines

We utilized the Scikit-Learn tools to train a Support Vector Machine (SVM) model for text categorization tasks. The code is evaluating the performance of the SVM model by utilizing the Unigram, Bigram, and Trigram text feature representations. A linear kernel SVM model is utilized for text classification.

Feature Extraction: The Unigram, Bigram, and Trigram n-gram formats are employed to extract characteristics from textual data.
Dataset Distribution: Along with the overall dataset, training set, and test set sizes, the dataset's distribution is detailed. Classification Report: A classification

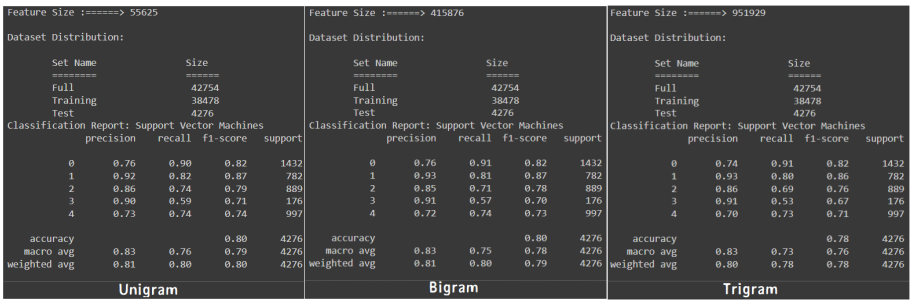


Figure 4.11: Support Vector Machines

analysis is provided for every one of the three n-gram representations. You may find measures like precision, recall, F1-score, and support for each class in the report, which ranges from 0 to 4. A thorough evaluation of the model's performance is achieved by computing the weighted average of the F1-score, as well as its precision and recall.

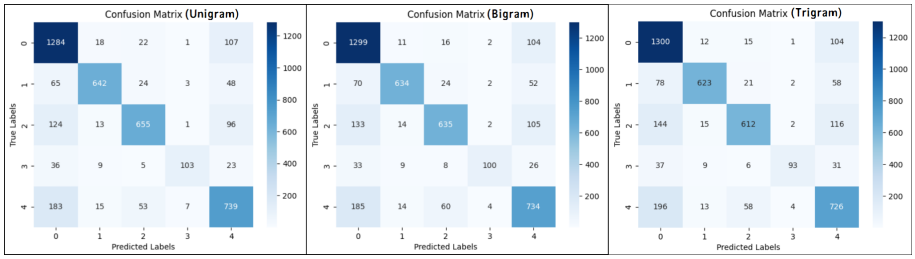


Figure 4.12: Support Vector Machines Confusion Matrix

Support Vector Machines (SVMs) were trained on Unigram, Bigram, and Trigram features. Overall accuracy ranged from 78% to 80%. Notable improvements in precision and recall were observed for Class 1. Feature size increased from Unigram to Trigram, impacting performance variations across classes.

4.1.7 Result Differentiation

The findings of an extensive analysis of several machine learning models for the detection of cyberbullying show that the XGBoost classifier consistently outperformed other models in terms of accuracy and general effectiveness. With an accuracy of more than 80%, the XGBoost model performed well across a variety of n-gram features, including Unigram, Bigram, and Trigram. The classification results for each class revealed high metrics for accuracy, recall, and F1-score, indicating the model's effectiveness in identifying instances of cyberbullying in social media posts. The key to XGBoost's success is its ensemble learning approach, which makes use of the abilities of several weak learners (decision trees) to create a powerful prediction model. XGBoost's flexibility allows it to deal with complex relationships in the data and adapt to different feature representations. The confusion matrices further support XGBoost's reliability in making accurate predictions across various classes.
The XGBoost model fared better than CatBoost, Multinomial Naive Bayes, Random Forest Classifier, Logistic Regression, and Support Vector Machines in terms of

cyberbullying detection. With an accuracy of almost 78%, CatBoost was effective; nevertheless, its performance differed depending on the feature set, and its length of training led to scaling issues, especially when the Trigram feature set was utilized. Multinomial Naive Bayes, a popular text classification algorithm, produced competitive results; nevertheless, it appeared that the choice of n-grams had a bigger influence on its overall performance. While Random Forest Classifier is an effective ensemble approach, its accuracy was similar to CatBoost rather than consistently matching XGBoost's performance. Even though Logistic Regression was straightforward to comprehend and apply, its accuracy was not as excellent as XGBoost's, implying that it is not as successful in dealing with complicated cyberbullying patterns. Support Vector Machines using a linear kernel generated competitive text classification results, but they were not as exact or precise as XGBoost. In general, the comparison study shows that XGBoost consistently outperforms the other models in terms of accuracy, precision, and overall performance. When it comes to discovering detailed patterns in a dataset, XGBoost's ensemble learning function looks to be really useful, making it an excellent pick.

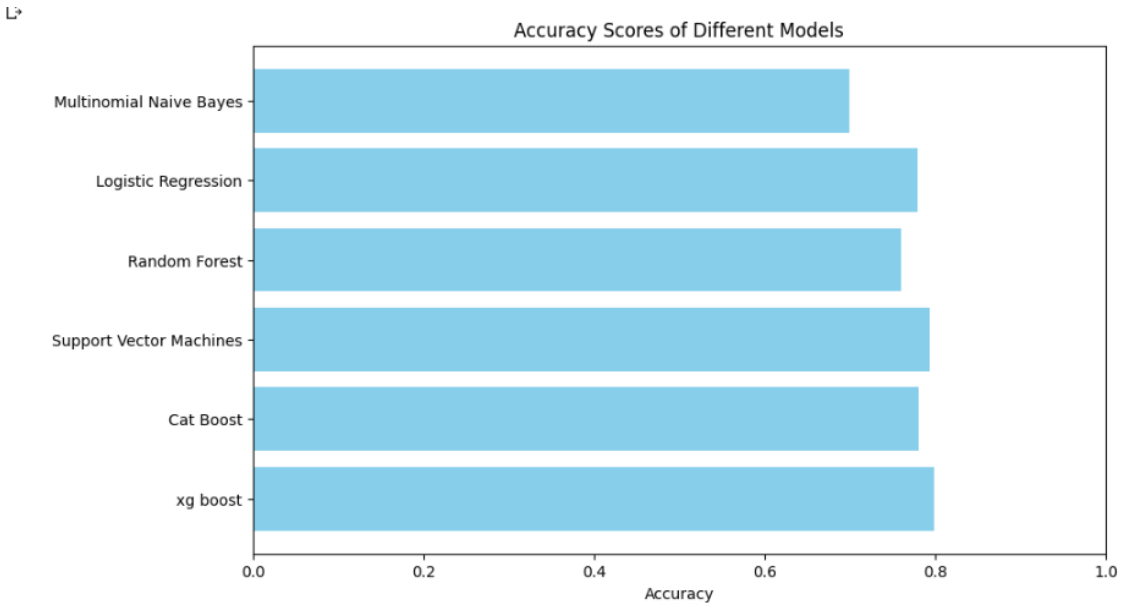


Figure 4.13: Differentiation in the accuracy of the Models Used

Chapter 5

Conclusion

The creation of a cyberbullying reporting system in Bengali is a key step toward making the internet a safer place for users who speak that language. We created Natural Language Processing (NLP) algorithms customized to the special elements of the Bengali language for significant distinguishing evidence and the reduction of cyberbullying occurrences in this specific semantic context. In order to establish a more complete and robust digital environment for Bengali speakers, this study underlines the need of responding to the different semantic requirements and obstacles in combating cyberbullying.

Bibliography

[1] D. Yin, Z. Xue, L. Hong, B. Davison, A. Edwards, and L. Edwards, *Detection of harassment on web 2.0*, Jan. 2009.

[2] O. Mendsaikhan, H. Hasegawa, Y. Yamaguchi, and H. Shimada, "Identification of cybersecurity specific content using the doc2vec language model," in *2019 IEEE 43rd Annual Computer Software and Applications Conference (COMPSAC)*, vol. 1, 2019, pp. 396–401. DOI: 10.1109/COMPSAC.2019.00064.

[3] H. Abie, "Cognitive cybersecurity for cps-iot enabled healthcare ecosystems," in *2019 13th International Symposium on Medical Information and Communication Technology (ISMICT)*, 2019, pp. 1–6. DOI: 10.1109/ISMICT.2019.8743670.

[4] Z. Tian, C. Luo, J. Qiu, X. Du, and M. Guizani, "A distributed deep learning system for web attack detection on edge devices," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 3, pp. 1963–1971, 2020. DOI: 10.1109/TII.2019.2938778.

[5] M. Tikhomirov, N. Loukachevitch, A. Sirotina, and B. Dobrov, "Using bert and augmentation in named entity recognition for cybersecurity domain," in *Natural Language Processing and Information Systems*, E. Métais, F. Meziane, H. Horacek, and P. Cimiano, Eds., Cham: Springer International Publishing, 2020, pp. 16–24, ISBN: 978-3-030-51310-8.

[6] M. A. Al-garadi, K. D. Varathan, and S. D. Ravana, *Cybercrime detection in online communications: The experimental case of cyberbullying detection in the twitter network*, 2016. DOI: https://doi.org/10.1016/j.chb.2016.05.051. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0747563216303788.

[7] K. Dinakar, R. Reichart, and H. Lieberman, "Modeling the detection of textual cyberbullying," Jan. 2011.

[8] R. Zhao, A. Zhou, and K. Mao, "Automatic detection of cyberbullying on social networks based on bullying features," in *Proceedings of the 17th International Conference on Distributed Computing and Networking*, ser. ICDCN '16, Singapore, Singapore: Association for Computing Machinery, 2016, ISBN: 9781450340328. DOI: 10.1145/2833312.2849567. [Online]. Available: https://doi.org/10.1145/2833312.2849567.

[9] M. F. Ahmed, Z. Mahmud, Z. T. Biash, A. A. N. Ryen, A. Hossain, and F. B. Ashraf, *Cyberbullying detection using deep neural network from social media comments in bangla language*, 2021. arXiv: 2106.04506 [cs.CL].

[10] C. Iwendi, G. Srivastava, S. Khan, and P. Reddy, *Cyberbullying detection solutions based on deep learning architectures*, Oct. 2020. DOI: 10.1007/s00530-020-00701-5.

[11] M. Tikhomirov, N. Loukachevitch, A. Sirotina, and B. Dobrov, "Using bert and augmentation in named entity recognition for cybersecurity domain," in Jun. 2020, pp. 16–24, ISBN: 978-3-030-51309-2. DOI: 10.1007/978-3-030-51310-8_2.

[12] H. Gasmi, J. Laval, and A. Bouras, *Information extraction of cybersecurity concepts: An lstm approach*, Sep. 2019. DOI: 10.3390/app9193945.

[13] M. Das Purba, B. Chu, and E. Al-Shaer, "From word embedding to cyber-phrase embedding: Comparison of processing cybersecurity texts," in *2020 IEEE International Conference on Intelligence and Security Informatics (ISI)*, 2020, pp. 1–6. DOI: 10.1109/ISI49825.2020.9280541.

[14] H.-S. Shin, H.-Y. Kwon, and S.-J. Ryu, *A new text classification model based on contrastive word embedding for detecting cybersecurity intelligence in twitter*, Sep. 2020. DOI: 10.3390/electronics9091527. [Online]. Available: http://dx.doi.org/10.3390/electronics9091527.