Mitigating Bias in Cyberbully Detection from Social Media with Text Classification

Zakaria Ibne Rafiq [21201357], Fahmid Hasan Chowdhury[21201286],
Radito Dhali[24141150], Rakibul Hasan Talukdar[23241100], Sabbir Hossain Mirza[23241086],
Amina Jannat Nurhan[23241099], Maliha Ishrar[21201434], Abdur Rahman Shafi[23241118]
BRAC University
Dhaka, Bangladesh

***Abstract*—With the rise of social media, cyberbullying has become a prevalent issue, causing significant emotional distress. Our model employs machine learning algorithms to categorize text based on demographic factors, including age, ethnicity, and religious beliefs. This project explores the use of machine learning for cyberbullying detection on Twitter. We train various algorithms such as Logistic Regression, Naive Bayes and K-nearest neighbour on a dataset of cyberbullying tweets. Our analysis reveals that Logistic Regression achieves the highest accuracy in identifying offensive language, highlighting the importance of selecting appropriate algorithms and preprocessing techniques for effective cyberbullying detection.**

***Index Terms*—Twitter, Naive Bayes, Logistic Regression, KNN**

I. Introduction

The rise of social media has brought a wave of connection, instant news, educational resources, and entertainment. However, this digital landscape also fosters negativity, with cyberbullying being a particularly harmful issue. Cyberbullying is the use of electronic communication to bully a person, typically by sending messages of an intimidating or threatening nature. It can manifest in a multitude of ways, ranging from sending cruel, hurtful, or threatening messages to spreading rumors or lies online. Cyberbullies may also share embarrassing photos or videos of someone without their consent, intentionally exclude them from online groups or activities, or even impersonate them to cause further distress. The consequences of cyberbullying for victims can be devastating, leading to anxiety, depression, social isolation, and even self-harm.

Cyberbullying has emerged as a serious threat alongside the rise of the digital world. A study recently published by Cyberbullying Research Center states that 34% of students reported experiencing cyberbullying in the past year.[1] Experiencing harassment online can affect a student in many bad aspects. Many projects and studies are held to tackle this problem using ML algorithms and NLP techniques. Our project hinges on a fundamental question: ”Does the cyberbullying detection system accurately detect harassing texts?”. Nowadays, some words labeled as bad words are used to express positive emotions. For example, ”filthy,” ”beast,” and ”wrecked” etc. can be flagged as cyberbullying indicators, but they can also have positive connotations. We aim to detect both generic cyberbullying texts and texts that might be easily misinterpreted as cyberbullying but aren’t actually bullying. This context-dependence is a major hurdle for cyberbullying detection systems, and it’s exactly what our project aims to address. We want to develop a system that goes beyond just identifying ”bad” words and can actually understand the intent behind the message.[2] We propose implementing our idea by considering four different aspects: age, gender, race, and ethnicity. We will then compare the performance of three different machine learning algorithms to identify the best one suited for our project.

Our proposed system strives to be an unbiased and valuable tool for detecting cyberbullying text on Twitter. This research not only contributes to creating a more positive online environment but also demonstrates the influence of data preprocessing and machine learning algorithm selection on model performance.

II. Literature Review

S. A. Mathur et al. [3] proposed a technique using NLP and ML algorithms to detect cyberbullying in real-time. They used the TF-IDF method to weight the word and feature selection. The author studied 4 ML algorithms- Random Forest Classifier, Gradient Boosting Classifier, AdaBoost Classifier and ran a test on a dataset of 47000 tweets collected from Kaggle which was balanced to ensure that each class had 8000 data. Results of their study showed that a tuned Random Forest Algorithm outperforms other ML algorithms in detection of cyberbullying tests.

While Bag-of-Words(BoW) achieved only 61.9% accuracy in sentiment and context analysis, projects like Ruminati as cited by S. A. Mathur et al. [3] using SVM for YouTube comments improved this to 66.7% by incorporating social factors and probabilistic modeling. This study proposes a deep learning network with a transformer architecture for cyberbullying detection. It utilizes a single neural network layer for classification. This model performs better on a larger dataset (Wikipedia) compared to a smaller one (Formspring), demonstrating increased accuracy and reliability with more data.

In a study by J.Wang et al.[4], eight vectorization techniques were compared: BoW, TF-IDF, word2vec, GloVe, fastText, BERT, DistilBERT, and sentence BERT. Their performance was evaluated using different machine learning algorithms on a dataset of 40000 tweets. The results showed that BoW achieved higher accuracy scores (0.8747-LR, 0.7876-NB, 0.5281-KNN, 0.8448-SVM, 0.9444-XGB, 0.8786-MLP) compared to TF-IDF (0.8514-LR, 0.8265-NB, 0.3086-KNN, 0.8095-SVM, 0.9386-XGB, 0.8375-MLP) for most machine learning algorithms. This suggests that for large datasets, BoW might outperform TF-IDF in certain tasks.

III. Methodology

A. Dataset

In our project, we use a dataset that is collected by [4] consisting of 48000 cyberbullying tweets based on age, gender, language, race, and ethnicity. The dataset used is collected for the purpose of fine-grained cyberbullying detection on social media platforms, specifically Twitter. It is generated through a semi-supervised online Dynamic Query Expansion (DQE) process, which extracts more natural data points of a specific class from Twitter. This method addresses the challenge of class imbalances present in existing cyberbullying datasets. Additionally, Graph Convolutional Network (GCN) classifier is used, which utilizes a graph constructed from thresholded cosine similarities between tweet embeddings. The dataset, augmented with DQE, is made publicly available to facilitate future research into fine-grained cyberbullying classification. This approach to dataset generation aims to provide a balanced multiclass dataset that enables a more representative sample of cyberbullying instances across the spectrum of offensive messages. This dataset gives us a variety of data to analyze and train our model on. With this rich dataset as fuel, our project is poised to make significant progress in developing a robust model for detecting cyberbullying across social media platforms. This advancement has the potential to create a safer online environment for everyone.

B. Preprocessing Text Data for Machine Learning

Preprocessing is a crucial step in preparing text data for machine learning tasks. Here, we'll explore two common techniques used in this process: text cleaning and feature extraction (TF-IDF).

The first step in text preprocessing involves cleaning the text data to ensure consistency and remove irrelevant information. This includes normalization, stop word removal, and stemming or lemmatization. Normalization converts all text to lowercase and removes punctuation marks, ensuring that "happy" and "Happy" are treated identically by the model.Stop words are common words like "the", "a", and "is" that carry little meaning. Natural Language Toolkit (NLTK) is a popular library that provides tools for stop word removal in various languages. Eliminating these words reduces noise in the data and allows the model to focus on more content-rich terms. Finally, stemming or lemmatization reduces words to their base form. For example, "running" becomes "run" (stemming) or "run"(lemmatization). This can improve efficiency and capture the semantic similarity between words. While stemming might create nonsensical words ("run" might not be the best root for "running" in all contexts), lemmatization aims to preserve the meaning of the word while reducing its variations. After applying these techniques, we might end up with a sentence like "tri send bible muslim moon saudi even allow country religion freedom israel." This cleaned text is then stored for further processing.

C. Feature Extraction with ***TF-IDF***

Machine learning models typically work with numerical data. To train our model on the cleaned text, we need to convert it into a numerical representation. Here, **TF-IDF** (Term Frequency-Inverse Document Frequency) comes into play.

$$TF = \frac{\text{Number of times a word "X" appears in a Document}}{\text{Number of words present in a Document}}$$

$$IDF = \log\left(\frac{\text{Number of Documents present in a Corpus}}{\text{Number of Documents where word "X" has appeared}}\right)$$

$$TF - IDF = TF \times IDF$$

$TF(t,d)-$ Term Frequency: This measures how often a term (word) t appears in a specific document d.
$IDF(t,D)-$ Inverse Document Frequency: This considers the overall importance of the term across all documents D in the dataset. Words that appear frequently across all documents will have a lower IDF weight, as they convey less specific meaning.
By multiplying these two values, **TF-IDF** assigns a weight to each word within a document. Words that are frequent in a specific document but rare overall receive a higher weight, indicating their significance for that particular document's topic.

This approach addresses the limitations of the Bag-of-Words model, which simply counts word occurrences without considering their importance within the broader context. TF-IDF helps us create a more nuanced representation of the text data, allowing the machine learning model to learn more effectively.

D. Machine Learning Algorithms

Naive Bayes is a powerful machine learning tool that excels in tasks like text classification and spam filtering. Despite its seemingly oversimplified assumption that features are independent, it leverages Bayes' theorem to achieve surprisingly accurate predictions. This efficiency comes from the fact that the algorithm can calculate the probability of a data point belonging to a specific class based on individual features, without needing to account for complex relationships between them. While the independence assumption may not always hold true, Naive Bayes often proves to be a fast and effective approach for various classification problems.

K-Nearest Neighbors (KNN) is a widely used technique for both classifying and predicting data. It's based on the principle that similar data points likely share similar labels or values.During training, KNN stores all the training data for reference. To make predictions, it calculates the distance between a new data point and all the training examples using a chosen metric like Euclidean distance. Next, KNN finds the K closest neighbors to the new data point. In classification tasks, it assigns the most frequent class label among these neighbors as the predicted label. For regression, it predicts the value by averaging or weight-averaging the target values of the K neighbors. KNN's simplicity and ease of understanding make it popular across various fields. However, its performance hinges on the chosen value of K and the distance metric. Therefore, careful parameter tuning is crucial for optimal results.

Logistic regression, a workhorse of machine learning, excels at classifying data points into distinct categories. Unlike linear regression for continuous values, logistic regression tackles binary classifications (spam/not spam) or multiple categories (malignant/benign tumors). The core mechanism lies in the sigmoid function, an S-shaped curve that transforms the linear relationship between features and the predicted probability. The output, between 0 and 1, reflects the chance of belonging to a specific class. Training refines the model. Logistic regression uses a cost function (often log-loss) to gauge the mismatch between predicted probabilities and actual labels. It iteratively adjusts feature weights to minimize this cost, essentially learning the interplay between features and the target category. Logistic regression shines with interpretability – feature weights reveal their importance in classification. It's also efficient and relatively simple to implement. However, it assumes a linear relationship between features and probability, and is limited to binary classification (extensions exist for multi-class).

Neural networks are a class of machine learning algorithms that excel at recognizing patterns in data just like our brain. These networks consist of interconnected nodes, mimicking neurons, that process information and learn through adjustments. The core of a neural network lies in its layered structure. Input data enters the network, travels through hidden layers performing calculations, and reaches the output layer with the final prediction. The hidden layers, the heart of the learning process, determine the network's ability to learn complex patterns.Neural networks learn through a process called back propagation. During training, the network receives data with known labels, makes a prediction, and compares it to the correct label. The difference between the prediction and the label is the error. The network then adjusts the connections between its nodes based on this error, in a way that minimizes the error for future predictions. Over many training iterations, these adjustments accumulate, allowing the network to learn and improve its performance.

IV. Results

A. Neural Network

Class	Precision	Recall	F1-score
Age	0.82	0.78	0.8
Ethnicity	0.82	0.83	0.86
Religion	0.78	0.83	0.8
Gender	0.72	0.7	0.71
Not-Cyberbyllying	0.84	0.85	0.84
Accuracy			0.82
Macro Avg.	0.81	0.8	0.8
Weighted Avg.	0.82	0.82	0.82

B. Logistic Regression

Class	Precision	Recall	F1-score
Age	0.94	0.88	0.91
Ethnicity	0.98	0.95	0.96
Religion	0.95	0.9	0.92
Gender	0.92	0.73	0.81
Not-Cyberbyllying	0.87	0.95	0.91
Accuracy			0.91
Macro Avg.	0.93	0.88	0.9
Weighted Avg.	0.91	0.91	0.91

C. Naive Bayes

Class	Precision	Recall	F1-score
Age	0.87	0.73	0.79
Ethnicity	0.88	0.83	0.85
Religion	0.82	0.89	0.85
Gender	0.86	0.64	0.73
Not-Cyberbyllying	0.81	0.89	0.85
Accuracy			0.83
Macro Avg.	0.85	0.79	0.82
Weighted Avg.	0.83	0.83	0.83

D. K- Nearest Neighbour

Class	Precision	Recall	F1-score
Age	0.73	0.94	0.82
Ethnicity	0.84	0.72	0.78
Religion	0.93	0.52	0.67
Gender	0.93	0.52	0.67
Not-Cyberbyllying	0.73	0.87	0.79
Accuracy			0.77
Macro Avg.	0.83	0.71	0.75
Weighted Avg.	0.8	0.77	0.76

Our evaluation using multiple metrics indicates that logistic regression emerged as the strongest performer for cyberbullying detection on your balanced dataset. It achieved the highest accuracy, signifying its ability to correctly classify both cyberbullying and non-cyberbullying instances. But the analysis goes beyond just overall accuracy. Logistic regression also excelled in precision and recall, both exceeding 90% on average. This means the model is highly accurate in identifying true cyberbullying cases (high recall) and minimizes false positives (high precision), avoiding wrongly accusing someone of cyberbullying. Consequently, the F1-score, which balances precision and recall, is also high. Finally, both macro and weighted averages show respectable scores, suggesting the model performs well across all classes, regardless of their frequency in the data. This is particularly important because it ensures the model's effectiveness is not skewed by the majority class. In conclusion, logistic regression demonstrates exceptional promise for accurately detecting cyberbullying in your balanced dataset.

V. Conclusion

This work contributes significantly to the fight against cyberbullying by developing a machine learning model that effectively classifies cyberbullying text. The model incorporates demographic factors like age, race, and gender,ethnicity offering a more nuanced understanding of online harassment. Notably, logistic regression outperformed other tested algorithms (KNN, Naive Bayes and Neural Network) in terms of accuracy, precision, recall, and F1-score. While KNN and Naive Bayes excelled in specific categories, logistic regression provided a more comprehensive solution.

Our project lays the groundwork for further exploration in cyberbullying detection. To enhance the accuracy of our model, we can address the following limitations:

Sarcasm and Irony Detection: Current models often struggle with nuanced language like sarcasm and irony, which can be misinterpreted as cyberbullying. Future work could involve incorporating techniques specifically designed to identify these forms of expression, leading to a more refined understanding of the true intent behind the communication.

Big Data Integration: The current model's performance might be limited by the amount of data it was trained on. Scaling the project to handle Big Data could significantly improve detection accuracy. However, we must acknowledge the computational cost associated with such an endeavor. Additionally, collecting and cleaning large datasets can be time-consuming and require careful planning.

Unsupervised Learning Approaches: While our initial work might have relied on supervised learning, exploring unsupervised learning techniques could be a valuable next step. This approach could help identify new patterns and trends in cyberbullying behavior that might not be readily apparent with supervised methods.

By addressing these limitations, the groundwork laid by this work paves the way for a more robust and versatile model. This will ultimately create a safer and more inclusive online environment for everyone.

References

[1] S. Hinduja, "Cyberbullying statistics 2021 — age, gender, sexual orientation, and race," 2021, accessed on: 2024-04-27. [Online]. Available: https://cyberbullying.org/cyberbullying-statistics-age-gender-sexual-orientation-race

[2] S. M. Kargutkar and V. Chitre, "A study of cyberbullying detection using machine learning techniques," in *2020 Fourth International Conference on Computing Methodologies and Communication (ICCMC)*, 2020, pp. 734–739.

[3] S. A. Mathur *et al.*, "Analysis of tweets for cyberbullying detection," in *2023 Third International Conference on Secure Cyber Computing and Communication (ICSCCC)*, 2023, pp. 269–274.

[4] J. Wang, K. Fu, and C.-T. Lu, "Sosnet: A graph convolutional network approach to fine-grained cyberbullying detection," in *2020 IEEE International Conference on Big Data (Big Data)*, 2020, pp. 1699–1708.