CS 334, Supervised by Dr. Joyce Ho, December 13, 2023

Evaluation of the Effectiveness of Different ML Models on Detecting Cyber Harassment

Andrew Chung, James Park

Emory University

1. Abstract

Cyberbullying, coined with other terms such as cyber harassment, is the use of technology and the internet to harass, threaten, embarrass, or target another person or group. With the rise of social media, games, and more people spending time on the internet in recent years, harassment on the web has reached an all time high and cyberbullying has become a significant issue that needs to be addressed. As concerned students ourselves, we wanted to approach this problem and investigate ways to prevent this act as much as possible. Thus, we drafted a plan to compare six different types of machine learning models and evaluate the effectiveness of these models in detecting cyber harassment.

2. Introduction

In 1997, the first recognizable social media site was established: Six Degrees. Although this website paved the path for posting pictures online and adding friends through the internet, social media was truly recognized as a means of communication after the release of Facebook in 2004. With individuals using the platform more frequently and it being incorporated into daily lives, new terms were eventually created to describe acts of harassment on the internet: cyberbullying, cyber harassment, trolling, and more.

Today, mostly affecting children and teenagers, cyberbullying affects countless students and brings worrisome effects, such as decreased academic performance, lack of self-esteem, and in severe cases, suicidal thoughts and suicide. Around 37% of middle and high school students have felt cyberbullied and there have been around 87% of reported observed cyber harassment cases; these numbers only continue to rise. Furthermore, these values have reached an all-time high after COVID-19 caused a world-wide lockdown, ultimately leading individuals to a more social-media and internet centered lifestyle and opening up more dangerous opportunities for cyberbullying. Our lives circle around technology, and we now live in an era where the internet rules us all. Therefore, it is imperative that we find ways to safely navigate these sites not only for our safety, but for those who we care about as well.

3. Background

To address this issue of cyber harassment, a study was done in 2015 by B. Sri Nandhini and J.I. Sheeba, where they used the FuzGen learning algorithm (uses the adaptive component of the system by means of a GA with fuzzy set genes) and Naïve classifier technique (a classification technique based on Bayes' Theorem with an independence assumption among predictors). The proposed framework consisted of data pre-processing, feature extraction (adjectives, nouns, verbs of similar topics), the FuzGen learning algorithm which models adaptive and exploratory behavior, and the Naïve Bayes Classifier for classifying bullying and non-bullying. Their proposed rules yielded high results: higher than 87% for accuracy, F1-measure, and recall. However, this study used text data from myspace.com and formspring.me, which are outdated websites. The way people communicate on the internet has changed over time; these models are trained on older diction that may not accurately depict classification of newer words.

Another study was done in 2021 by Aditya Desai, Shashank Kalaskar, Omkar Kumbhar, and Rashmi Dhumal, where they used a dataset from Twitter, now X, to evaluate sentiment analysis and cyberbullying classification on different ML models. Their proposed framework consisted of using SVM, Naïve Bayes, and BERT-base models. Text vectors were created through TF-IDF. An input would be fed to the BERT-base model, which will then be classified by SVM and Naïve Bayes, which will provide an output, being a detected sentiment and further classifying whether a text is cyberbullying.

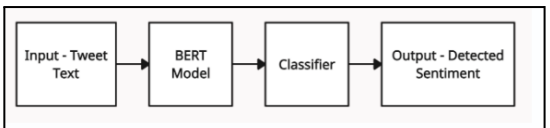


Fig.2. BERT model flow chart based on sentiment analysis

Their pre-trained BERT model held an accuracy of 70.89% and 91.9% for testing and training sets, respectively. Although this provides high results, we are looking to classify with different techniques. Firstly, we will be using word-embedding (Word2Vec) instead of TF-IDF, as we are looking to understand meaning and context of words in the dataset instead of identifying the relevancy of the words. Furthermore, we will be using text classification instead of sentiment analysis as we are looking for a simple binary output (either cyberbullying or non-cyberbullying) instead of specifying what type of cyberbullying it is. Lastly, we are planning to use K-Nearest Neighbors, Decision Tree, Logistic Regression, and more transformer based models (different variations of the BERT-base model). Our approaches for this issue are different, and thus, can yield different and interesting results.

---

4. Methods

Since we are looking to classify cyberbullying and non-cyberbullying in a binary fashion, we looked into six ML models appropriate for this study: K-Nearest Neighbors, Decision Tree, Logistic Regression, BERT-base, RoBERTa-base, TwHIN-BERT-base.

4.1 K-Nearest Neighbors

K-Nearest Neighbors, often abbreviated as KNN, is a non-parametric supervised learning model that classifies or predicts

an individual point or a group of points based on proximity to other points in the dataset. The name is derived from choosing a value *k,* such that *k* depicts the number of nearby values that would be taken into consideration when classifying.

For a binary classification problem like this one, KNN would look into nearby points and increment the respective counts of classifications. Afterwards, KNN would predict that the point has the same classification as the majority group. Due to this, it is important to select an odd value for *k,* as this can break ties when counts are equal.

4.2 Decision Tree

Decision Tree is also a non-parametric supervised learning model used for classification and regression, where rules are created from different features of the dataset to create a model of decisions and their possible consequences, including chance event outcomes, resource costs, and utility. The model's hyperparameters include minimum leaf samples and maximum depth; with these two combinations, an ideal size for a decision tree can be found to most accurately perform binary classification.

4.3 Logistic Regression

Logistic Regression is a supervised learning model that makes use of logistic functions to predict the probability of a binary outcome. It computes a sum of the input features (in most cases, there is a bias term), and calculates the logistic of the result. Since the output of a logistic regression is between 0 and 1, it is suitable for a binary classification task. The model's hyperparameters include solvers, penalties, learning rate (C), class weights, and max iterations.

4.4 BERT-base

BERT (Devlin et al., 2019), is a pre-trained language model based on Transformer architecture (Vaswani et al., 2017). The model is pre-trained on a large corpus and is able to be fine-tuned for many different types of tasks, such as semantic analysis, question answering, named entity recognition, and many other common NLP tasks. BERT-base is the smaller of the two BERT models, but both models are trained in the same way.

Rather than training a completely new model from scratch, BERT is a model that has already been pre-trained on the Book corpus, as well as Wikipedia. It utilizes Masked Language Modelling and Next Sentence Prediction to gain deep understanding of relationships between words and sentences, allowing it to do a multitude of tasks. To fit to specific tasks wanted by the user, the model solely needs to be fine-tuned in order to fit the requirements of the problem that one is trying to solve.

4.5 RoBERTa-base

RoBERTa (Liu et al., 2018), is a pre-trained language model that is an improved version of the original BERT model. It was trained on a much larger data corpus, amassing over 160GB of data. It also trains without the usage of NSP and utilizes a dynamic version of MLM, training on longer sequences and more epochs. In most cases, RoBERTa tends to outperform BERT on many common NLP

tasks. The overall usage of RoBERTa stays similar to BERT, as it is also a large-pretrained model that needs to be fine-tuned rather than being trained from scratch.

4.6 TwHIN-BERT-base

TwHIN-BERT-base (Zhang et al. 2018) is a BERT model that has been pre-trained on the Twitter corpus, and functionally is not different from BERT, as it follows the same model architecture.

---

5. Experiments/Results

To answer the question which model detects cyberbullying the best, we needed to find a dataset that we could use, preprocess this data, and run classification tests on this data to compare the results.

5.1 Data Description

"Cyberbullying Classification" from Kaggle is a dataset of 46,017 unique tweet texts scraped from twitter.com. The dataset originates from J. Wang, K. Fu, and C.T. Lu's paper "SOSNet: A Graph Convolutional Network Approach to Fine-Grained Cyberbullying Detection." There are two main categories, the first column being the tweet itself, and the second column being the classification of the tweet as not cyberbullying or the type of cyberbullying: age, religion, gender, ethnicity, other cyberbullying. The tweets were classified by humans. For this project, we will need to first preprocess the text data, as it is currently untokenized and some contain unnecessary parts such as the usernames or other outlying punctuation. We will also need to change the types to solely be two types, cyberbullying and not cyberbullying, as we are just interested in classifying messages as either or. The original dataset was edited so that there would be an equal amount of cyberbullying and non-cyberbullying examples (now approximately 15,000 entries); random cyberbullying examples were removed to match the count of the other. We deemed that this would be fine in this case, as we are not classifying the type of cyberbullying.

5.2 Preprocessing

Since we used text data, preprocessing involved various steps: standardization, tokenization, word-embedding, and binary classification. Standardization involves altering the text so that the models can easily interpret the data: removing NA values, stop words, special characters, and lowercasing all words. We then split the text into individual tokens using the The Natural Language Toolkit, more commonly known as NLTK, a suite of libraries and programs for symbolic and statistical natural language processing. We used the Word2Vec method for generating word embeddings from training data (an 80% split from the full set). We thought that Word2Vec embeddings would be better than GloVe, another traditional word embedding method for semantic analysis, as Word2Vec training is often faster and requires less memory compared to GloVe; it is more efficient, especially for smaller datasets. Given the classification column, we also classified not cyberbullying and cyberbullying as binary values (0 and 1, respectively) for easier analysis and recording when using the different model types.

5.3 Modeling Choices

KNN, Logistic Regression, and the Decision Tree models were specifically chosen as the goal of this experiment was to detect if a tweet was considered to be cyber harassment or not, and these models are utilized in many different binary classification tasks. The three transformer based models, BERT-base, RoBERTa-base, and TwHIN-BERT-base were chosen as these models are often utilized in NLP tasks, and in this case were used to understand tweets and classify them in relation to cyber harassment. To train the models, random hyperparameters were initially chosen, and were later fine-tuned for the KNN, Decision Tree, Logistic Regression models. For the transformer models, we deemed that hyperparameter tuning was not necessary, as the models reached efficient performances with just the initial parameters. After finding the optimal hyperparameters for KNN, Decision Tree, and Logistic Regression, we tested each model utilizing F1, precision, recall, and accuracy scores, comparing results to each other. By far, the best performing models were the three transformer models.

KNN:

Hyperparameter Tuning:

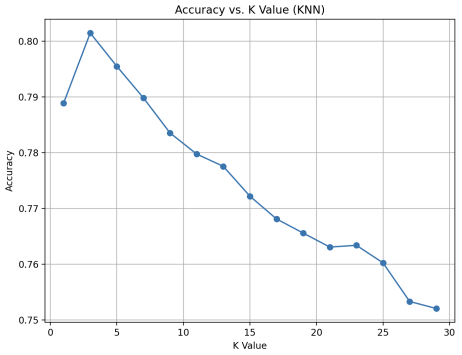
param_grid =

'n_neighbors': range(1, 31, 2),

'weights': [‘uniform’, ‘distance’]

Best Hyperparameters:

{'n_neighbors’: 3, 'weights': distance}



Decision Tree:

Hyperparameter Tuning:

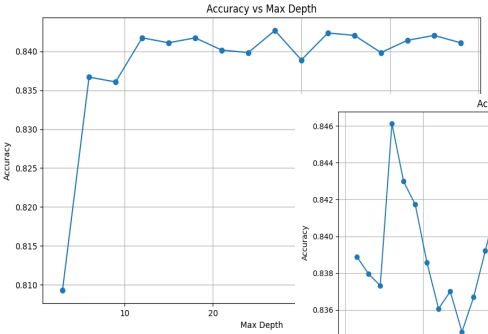
param_grid =

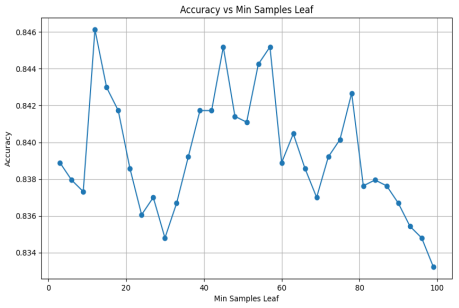
'min_samples_leaf': range(1, 101),

'max_depth': range(1, 101)

Best Hyperparameters:

{'max_depth': 27, 'min_samples_leaf': 12}





Logistic Regression:

Hyperparameter Tuning:

param_grid =

'C': [0.001, 0.01, 0.1, 1, 10, 100],

'penalty': ['l1', 'l2'],

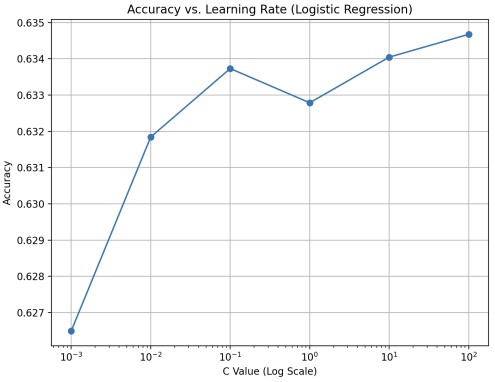
'solver': ['liblinear', 'newton-cg', 'lbfgs', 'sag', 'saga'],

'class_weight': [None, 'balanced'],

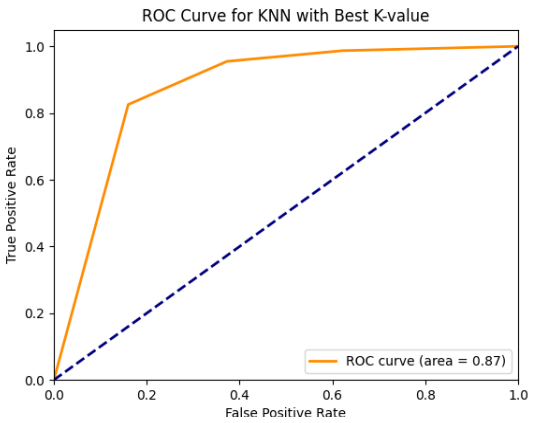
'max_iter': [100]

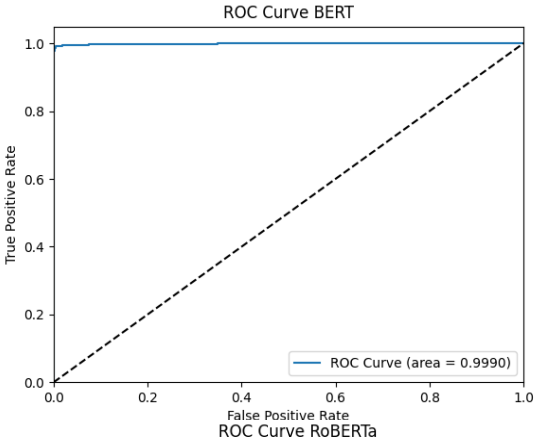
Best Hyperparameters:

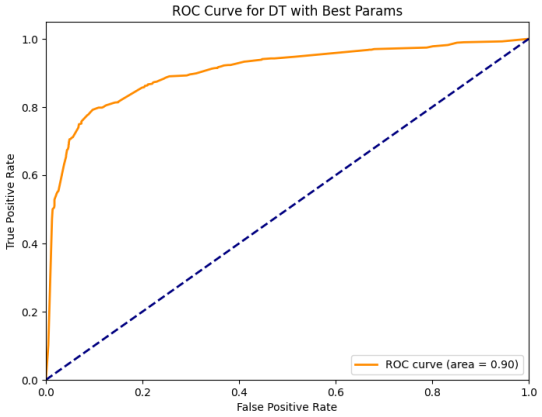
{'C': 100, 'class_weight': None, 'max_iter': 100, 'penalty': 'l2', 'solver': 'liblinear'}

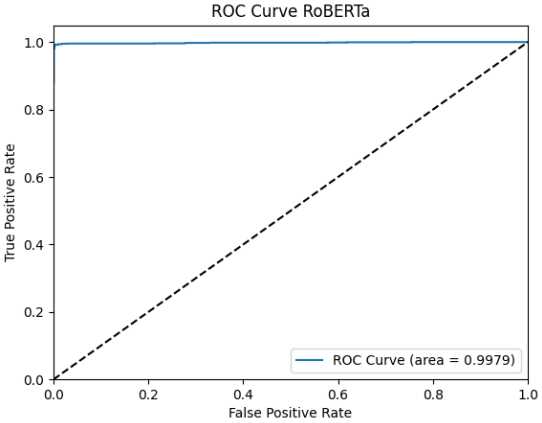


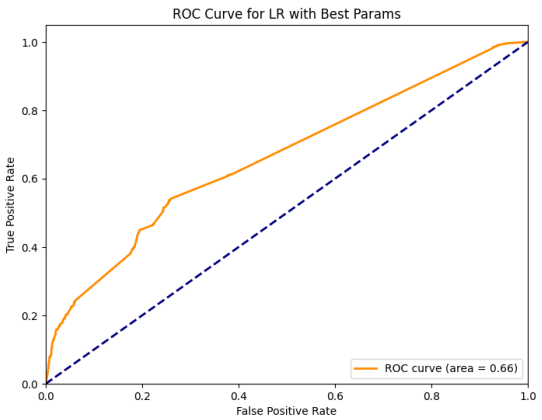
5.4 Empirical Results

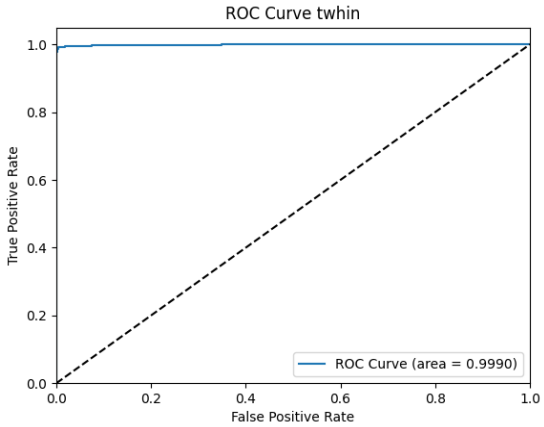












In order to evaluate the models, we utilized accuracy, precision, recall, and F1 scores for accurate and clear comparison. As we utilized a dataset with equal division of each of the classifications, accuracy was used as a measure of model performance. We also utilized F1 scores to get perspective on incorrectly classified classes. Precision was used to check how correct the model was when predicting the target class. Recall was used to show whether a model could find all objects of the target class. AUROC was utilized to visualize the performance of the models based on the rate of accuracy in relation to the effectiveness of the binary

classification. For simplicity, we show F1 score (representing precision and recall), accuracy, and AUROC scores in the table below:

Model	Accuracy	F1 score	AUROC
KNN	79%	74%/82%	0.87
DT	85%	85%/84%	0.90
LR	64%	67%/59%	0.66
BERT	99%	99%/99%	0.99
RoBERTa	99%	99%/99%	0.99
TwHIN	99%	99%/99%	0.99

Figure 1. Table representing the accuracy, F1 score for both of the classes (no cyberbullying on the left, cyberbullying on the right) and the AUROC score. This was utilized for performance evaluation.

From the six models, it is evident that the three transformer models vastly outperform the other three other classification models. The transformer models reach an accuracy of 99% respectively, with Decision Tree and KNN falling slightly short at 85% and 79% accuracy respectively. By far the worst performing model was Logistic Regression, giving both the worst accuracy and F1 scores of all six models. In general, the transformer models give the most consistent and accurate results from when tested on the Kaggle dataset.

6. Discussion

The three transformer models have performed at a very accurate level, reaching 99% for all metrics we have utilized to test the models. We believe this is because the task at hand is rather simple for such complex models that have been trained on large amounts of data. The pre-trained models have a deep understanding of words, sentences, and relationships between phrases, leading to efficient and accurate classification. Even as we searched for faults in our training methods and any sign of data leakage, there was no error that we could find.

In the future, we would like to classify the types of cyberbullying, as this may lead to more fine-grained and accurate results in terms of usage of this model outside of testing with just a dataset. While this is a much more complex task leading to initial drops in accuracy and model understanding, it may lead to more practical use cases and better model understanding and training overtime.

Also, figuring out a way to differentiate between a “negative” or “critical” comment and actual cyber harassment is something we would look into in the future. The subtle differences in these types of comments seemingly are hard to distinguish more from the model. While it is obviously better to have false positives in terms of harassment, the model would have much more practical use cases if it was able to distinguish between these types of comments in a more accurate way.

Lastly, another interesting approach, similar to the study done in 2021 mentioned in our background (Section 3), would be to train these models to test for semantic analysis instead of simply text classification. This would allow us to not only determine whether or not if a given text is considered cyberbullying, but can further look into emotions and build a deeper understanding. Furthermore, we used additional models that can be tested.

7. Contributions

Andrew Chung: Responsible for training the three transformer models, as well as testing the model performance for these three models. He wrote the model explanations for the transformer models, as well as part of the experiments/results and discussion section of the paper.

James Park: Responsible for training KNN, Decision Tree, and Logistic Regression, as well as testing the model performance for these three models. He wrote the model explanations for KNN, Decision Tree, and Logistic Regression, as well as part of the experiments/results, abstract, introduction, and background section of the paper.

---

8. Code/Dataset

The following leads to the GitHub where all code, dataset, and results are kept:
https://github.com/AChung1020/mlFinalProject