Offensive Language Identification System

Anindya Sen[1], Sugam Jaiswal[2], Pranshu Acharya[3], Dr. Jaisakthi S M[4]

[1-4]Vellore Institute of Technology, Department of Computer Science, Vellore, Tamil Nadu, India

Abstract - *In this paper, we propose a system for OffensEval (SemEval 2019 Task 6), which involves offensive language to be recognized. When a user provides input, the paper's goal is to detect objectionable statements in that input. A text or a file can be submitted by the users, depending on the size of their inputs and their convenience in providing such information. This is accomplished through the use of a web application that serves as a platform for receiving user inputs and comparing them to the suggested model in order to detect offensive statements included within them. Natural language processing techniques are utilized to clean up the input data, and machine learning algorithms are employed to detect sentences that are considered objectionable to the target audience. In order to detect objectionable language, the model is trained using a dataset that is given by the competition.*

***Key Words*:** Web Application, Machine Learning Algorithms, Natural Language Processing, Flask.

1.INTRODUCTION

The paper intends to propose a solution for offensive language problem by providing a web application that allows users to check their inputs for offensive nature. Users will be able to give their inputs in the form of text or file. Text format can be used if they have specific statements to check, while file format can be used if there is large number of sentences for example emails, assignments, documents, presentations. There are many models and algorithms that can be used to identify offensive statements, but the preferred model would be that one which is not only accurate, but also fast. Time is considered because we don't want users to get impatient by waiting for a long time to get their results.

1.1 Problem Statement

The usage of offensive language by people has been rising since the past few decades and is now at its peak. With the increasing number of social media, people can simply sit at their homes and use strong language without the fear of being caught. This has resulted in many of the social medias simply blocking or banning such users from using their social sites. However, people can make multiple accounts and continue with their hazardous behavior, which not only affects a particular person or a group that is being targeted, but also the other users who read such comments and opinions. This problem does not only occur in social medias, but has been showing a repeating trend in work emails, social websites, multiplayer games, and a lot more places. Most people can understand such language and ultimately take certain actions against it, but when you consider the entire world, not everyone can understand each other's language and that is how a lot of people can get targeted to this abuse.

A typical problem that occurs when dealing with offensive statements is that you have to read them. This can cause multiple emotions to rise such as anger, frustration, disappointment. Another type of case that is also seen quite often is that people don't know or understand if a particular statement is offensive or not. Thus, they unintentionally send offensive statements to their friends, family, colleagues, etc, which puts them in situations they don't want to be in. Thus, there is a need of a tool that can not only be used by companies like social media, but also by individuals to determine if some text or document has any offensive statements or not.

1.2 Motivation

Expansion in the utilization of social media locales likes Facebook and Twitter have given the group an incredible stage to offer their viewpoints/affections for the individual, gatherings or occasions occurring around them or in the public eye. This advanced media has turned into an extraordinary asset to share the data and furthermore gives the full ability to speak freely to everybody on the stage. With the acquiring prominence of these stages, there likewise comes the negative part alongside its benefits. This element of the web-based media to communicate something straightforwardly to the world have made the serious issues for these online organizations and adversely affected the prosperity of the cultural dignity. There are expanding instances of the maltreatment or offense on the online media like Hate discourse, Cyber-harassing, Aggression, or general Profanity. It is particularly essential to comprehend that this conduct cannot just monstrously affect the existence of an individual or a gathering however could be self-destructive sometimes; antagonistically hampering the psychological wellness of the person in question/s.

This expanding negative circumstance on the web has provoked a gigantic interest for these online media stages to embrace the assignment of identifying the questionable substance and making the proper move which can forestall the circumstance turning out to be more awful. This errand of identifying the offensive substance can be played out my human mediators physically, however it is both for all intents and purposes infeasible just as tedious considering the measure of the information produced on these online media stages, thusly there is a need to fill this hole. considering the significance and the affectability of this specific subject in the present advanced world there is, still a ton of arising further extension in handling and make do on the past work done in the field with the assistance of these new age AI and NLP strategies.

Posting offensive or harmful language via web-based media have been a genuine worry as of late. This has made a ton of issues in view of the colossal ubiquity and use of online media destinations like Facebook and Twitter. The fundamental motivation lies in the way that our model will mechanize and speed up the identification of the offensive language to work with the significant activities and control of these offensive posts.

2. Literature Review

[1] The authors used a text- based CNN Model to classify the various offensive languages. The model was implemented by using 2 CNNs for classification and sentiment analysis. CNN 1 is an improved version of CNN, whereas CNN 2 is a combination of multiple CNNs. The dataset being used is from the official competition, OLID, which was analysed using simple NLP coding. The paper not only finds the offensive language but also classifies them as Individual, Group or other.

[2] The authors have used a combination of Convolutional Neural Network, Bidirectional LSTM The different architectures mentioned are a part of a deep learning model. This model allowed the team to identify and categorize the tweets as targeted or not, to a particular audience. Additional dataset was obtained through Twitter API by searching for tweets containing selected keyword patterns. The model proposes a fast and fairly accurate method (deep learning) to classify the tweets. The model uses various keywords to search for relevant content and there is quite a possibility to miss out a lot of tweets.

[3] The authors have used supervised machine learning algorithms for identifying and classifying of offensive text. One of the important algorithms used is the SVM algorithm. They have taken advantage of the simple model of word embeddings to obtain vector representation of words for processing. The dataset used here was the one provided by the competition, which contained a total of 13240 training and 860 testing sentences. The model is very fast and simple to implement as it uses only machine learning algorithms. The team achieved third rank in one of the subtasks. The class distribution is highly imbalanced due to which there might be a bias introduced by the training algorithms.

[4] The paper describes the results and main findings of the competition. It states that deep learning was the most popular approach from majority of the teams and that it was used by about 70% of the teams. The dataset obtained consists of 14000 English tweets which were collected by 2 members. BERT was the best model for identification and target identification whereas Rule-based approach was the best model for knowing if it was targeted or not. The paper does mention popular approaches and best approaches but fails to explain other approaches that could have worked even better, or the authors' suggestions on how they could have improved.

[5] The authors used an ensemble of several models (LSTM, Transformer, SVM, Random Forest) for the classification. The main models used from neural networks are LSTM and (BERT) Transformer, from machine learning are Random Forest and SVM. The also computed the Tf-Idf score with various features to make the classifier. The paper explains how the authors pre-processed the dataset. It includes removing usernames, removing links and other unwanted texts. The classifier approach used by them is usable for all the tasks from the competition. It allows for multiple classification using the same model. The model does provide fairly accurate results but at the cost of the time. Using so many algorithms slow down the time required to process significantly.

3. Methodology

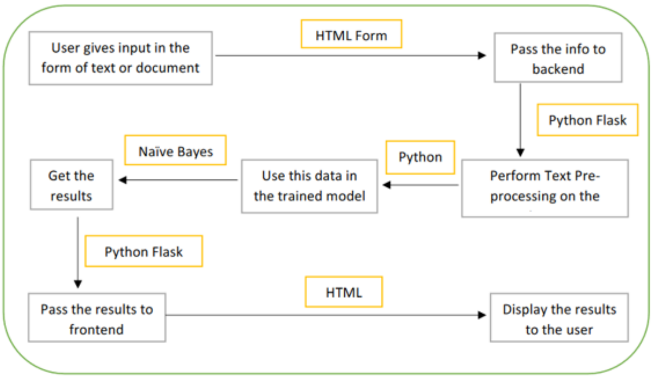

User gives input in the form of text or document
HTML Form
Pass the info to backend
Python Flask
Naïve Bayes
Python
Get the results
Use this data in the trained model
Perform Text Pre-processing on the
Python Flask
HTML
Pass the results to frontend
Display the results to the user

Fig -1: Proposed model

3.1 Offensive language Identification

This is the part of the work that deals with identifying the offensive user inputs. The script is written in Python and uses a Naive Bayes Classifier which is a Machine Learning model to train and test the model.

Firstly, we must pre-process the text to make it usable in the form required for the machine learning algorithm. In text pre-processing, we have removed unwanted characters such as @, #, stop-words, emojis, punctuations and digits. We will also be lemmatizing the data, that is converting it to basic form (dancing, dances, etc. -> dance).

After that label encoder is used to store all unique words and assign a unique numeric value to each of them. This is used in the Tf-Idf vectorizer to create Tf-Idf of the bag of words which will be used in the Naïve-Bayes, SVM Model and LSTM to recognize likeliness of words to be classified as offensive. All these models use the term frequency matrix that has been calculated before and will predict the probability of the text to be classified as offensive or not.

3.2 Web Application

The front-end webpage is designed using HTML, CSS and JavaScript. The back end, which contains the main script for running the model that will be used for offensive identification, is done using Python. They are connected using Python Flask which is a microframework written in Python.

3.3 Dataset

The dataset used in this paper is provided in the official SemEval competition, called Offensive Language Identification Dataset (OLID). It contains a total of 13240 tweets from twitter labelled as offensive or not. Three-fourths of the tweets are used for training and the remaining are used for testing the accuracy of the model.

3.4 Text Classification

After text pre-processing, the dataset is divided into two parts, training and testing. Training dataset will contain 3/4th of the dataset that is 9930 data items, whereas the Testing dataset, which will be used to test the accuracy of the models, will contain the remaining part of the dataset mainly, 1/4th of the dataset which is 3310 data items.

Also, the part of the dataset which tells if the text before it is offensive or not is converted into either 0 or 1 (0 -> Not Offensive, 1 -> Offensive) to make it easier for training and testing purposes. This is done using an Encoder function.

Also, we will be using Tf-Idf Vectorizer to replace the unique words in the dataset with unique numbers to represent them.

4. Model Training and Testing

4.1 Naïve Bayes Model

It is a classification technique based on Bayes' Theorem with an assumption of independence among predictors. In simple terms, a Naive Bayes classifier assumes that the presence of a particular feature in a class is unrelated to the presence of any other feature.

The Naïve Bayes Model gives an accuracy of around 71%, but only takes 0.01 seconds to train and test the model.

4.2 Support Vector Machines (SVM)

A support vector machine (SVM) is a supervised machine learning model that uses classification algorithms for two-group classification problems. After giving an SVM model sets of labeled training data for each category, they're able to categorize new text.

The SVM Model gives an accuracy of 77%, but takes 23 seconds to train and the test the model, which has almost 6% more accuracy at the cost of additional 23 seconds which is more than 2000 times slower than the Naïve Bayes Model

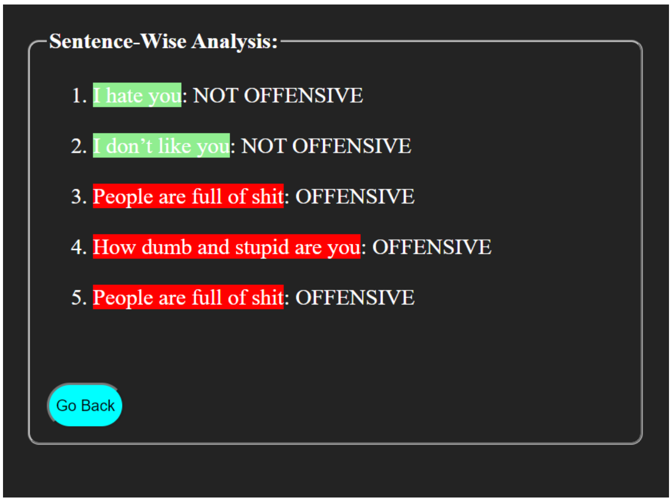
4.3 Long Short-Term Memory (LSTM)

Long short-term memory is an artificial recurrent neural network architecture used in the field of deep learning. Unlike standard feedforward neural networks, LSTM has feedback connections. It can process not only single data points, but also entire sequences of data.

The LSTM Model gives an accuracy of almost 75%, but takes 0.28 seconds to train and the test the model, which has almost 5% more accuracy at the cost of additional 0.27 seconds which is 280 times slower than the Naïve Bayes Model.

The Naïve Bayes Classifier is very fast and suitable for the web-application. Even though SVM, LSTM models are respectively 6%, 5% more accurate, they are still very slow and to make the user wait for long amounts of time will make the application look weak. Thus, we have used the Naïve Bayes Classifier for the offensive text identification.

5. Results and Discussion



Sentence-Wise Analysis:

1. I hate you: NOT OFFENSIVE
2. I don't like you: NOT OFFENSIVE
3. People are full of shit: OFFENSIVE
4. How dumb and stupid are you: OFFENSIVE
5. People are full of shit: OFFENSIVE

Fig -2: Output

5.1 Performance Analysis

This paper has used three different models for offensive language identification.

The Multinomial Naïve Bayes Model which has an accuracy score of 71%.

The SVM Model works with an accuracy of 77%.

The LSTM Model works with an accuracy of 75%.

Table -1: Comparison of the proposed models

Research Paper	Model Used	Accuracy Score
NLP@UIOWA at SemEval-2019 Task 6: Classifying the	Multi-Windowed CNNS	78%

Crass using Multi- windowed CNNs		
MIDAS at SemEval-2019 Task 6: Identifying Offensive Posts and Targeted Offense from Twitter	Ensemble of CNN, BLSTM with Attention, BLSTM + BGRU	84%
Fermi at SemEval-2019 Task 6: Identifying and Categorizing Offensive Language in Social Media using Sentence Embeddings	ELMo Sentence Embeddings with SVM Classifier using RBF Kernel	78.2%
NLPR@SRPOL at SemEval-2019 Task 6 and Task 5: Linguistically enhanced deep learning offensive sentence classifier	Neural Models such as LSTM and Transformer Models mixed with Machine Learning Random Forest and SVM	80%
NULI at SemEval-2019 Task 6: Transfer Learning for Offensive Language Detection using Bidirectional Transformers	Bidirectional Transformers, BERT	85%
Nikolov-Radivchev at SemEval-2019 Task 6: Offensive Tweet Classification with BERT and Ensembles	BERT-Large	85.5%
CAMsterdam at SemEval- 2019 Task 6: Neural and graph-based feature extraction for the identification of offensive tweets	Feature Extraction from RNN with ELMo embeddings	84.5%

5.2 Limitations

As mentioned before, one of the limitations of the paper is its accuracy. Even though the application is fast, it still could give inaccurate results that defeats the whole purpose of this paper. Moreover, offensiveness is a subjective topic. What seems offensive to one person, may not be to every other person, and therefore it is difficult to actually classify something as offensive or not.

Therefore, there is a high chance that a user may not be satisfied with the results as their definition or limit of offensiveness may be different from what the model is trained to recognize. Another limitation is that the file input can only be used by the model if it is in .txt or .doc format. There are tons of other types of file which the model cannot process.

5.3 Future Work

This paper is not perfect and can use some modifications/additions to make it function even better. Firstly, a better algorithm can be developed and used for the offensive language identification, that provides a balanced result when considering factors like accuracy and speed. Secondly, the front-end of the application is user- friendly, but still needs improvements. Only CSS is not enough for making the webpages attractive and thus, more tools such as bootstrap can be used to attract more people to use the application.

6. Conclusions

Offensive language identifiers are being used by a lot of social media sites to block and prevent the use of such language. However, most of them only block the abusive words and not the offensive texts which is inefficient in a lot of cases as simply blocking the abusive words would not solve the problem. Moreover, such technology and tools are only accessible to the companies that use them, and there is no similar platform or application that can be used by individuals to do the same. Even though there a lot of codes that can be used by the people for this purpose, no one would want to take the effort of search up codes, running them and then using it with their text.

This paper would provide the people, a tool where they could check their texts and documents without having to worry about coding. Such functionality is still not available on the internet, and thus we feel that our outcome is unique.

REFERENCES

[1] Rusert, J., & Srinivasan, P. (2019, June). NLP@ UIOWA at SemEval-2019 Task 6: Classifying the Crass using Multi-windowed CNNs. In Proceedings of the 13th International Workshop on Semantic Evaluation (pp. 704-711).

[2] Zhang, H., Mahata, D., Shahid, S., Mehnaz, L., Anand, S., Singla, Y., ... & Uppal, K. (2019). Identifying offensive

posts and targeted offense from twitter. arXiv preprint arXiv:1904.09072.

[3] Oswal, N. (2021). Identifying and Categorizing Offensive Language in Social Media. arXiv preprint arXiv:2104.04871.

[4] Seganti, A., Sobol, H., Orlova, I., Kim, H., Staniszewski, J., Krumholc, T., & Koziel, K. (2019). NLPR@ SRPOL at SemEval-2019 Task 6 and Task 5: Linguistically enhanced deep learning offensive sentence classifier. arXiv preprint arXiv:1904.05152.

[5] Liu, P., Li, W., & Zou, L. (2019, June). NULI at SemEval-2019 task 6: Transfer learning for offensive language detection using bidirectional transformers. In Proceedings of the 13th international workshop on semantic evaluation (pp. 87-91).

[6] Nikolov, A., & Radivchev, V. (2019, June). Nikolov-radivchev at semeval-2019 task 6: Offensive tweet classification with bert and ensembles. In Proceedings of the 13th International Workshop on Semantic Evaluation (pp. 691-695).

[7] Aglionby, G., Davis, C., Mishra, P., Caines, A., Giannakoudaki, H. Y., Rei, M., ... & Buttery, P. (2019). CAMsterdam at SemEval-2019 Task 6: Neural and graph-based feature extraction for the identification of offensive tweets.

[8] Hu, Y., Li, Y., Yang, T., & Pan, Q. (2018, November). Short text classification with a convolutional neural networks based method. In 2018 15th International Conference on Control, Automation, Robotics and Vision (ICARCV) (pp. 1432-1435). IEEE.

[9] Aglionby, G., Davis, C., Mishra, P., Caines, A., Giannakoudaki, H. Y., Rei, M., ... & Buttery, P. (2019). CAMsterdam at SemEval-2019 Task 6: Neural and graph-based feature extraction for the identification of offensive tweets.

[10] Zhang, H., Mahata, D., Shahid, S., Mehnaz, L., Anand, S., Singla, Y., ... & Uppal, K. (2019). Identifying offensive posts and targeted offense from twitter. arXiv preprint arXiv:1904.09072.

[11] Oswal, N. (2021). Identifying and Categorizing Offensive Language in Social Media. arXiv preprint arXiv:2104.04871.

[12] Nikolov, A., & Radivchev, V. (2019, June). Nikolov-radivchev at semeval-2019 task 6: Offensive tweet classification with bert and ensembles. In Proceedings of the 13th International Workshop on Semantic Evaluation (pp. 691-695).