Bully Comment Flagger

University of Washington - TCSS 554
Aaron Devlin, Lan Ly, Alex Pawlak, Jowy Tran
12/12/2017

Problem Description

The widespread use of social media creates an environment that often tempts certain users to become cyber bullies. Physical bullies usually deal with the police, but the majority of cyber bullies are cloaked with anonymity, making it easier to post as they please with few repercussions. These bullies generally do not know how their actions affect the bullied party, which can amount to fatal consequences in many cases. The target user is anyone that visits social media sites, or uses a text box to interact with other users. This implementation of bully comment flagging can identify a comment as a bully comment by analyzing the message before it is sent on Facebook or Twitter, for example. If the comment is not-questionable, then it is posted normally. When the comment is found to be a bully comment based on the analyzer, the poster is notified. The notification informs the user that his or her message is bullying and is not appropriate to post.

This can easily be implemented as a feature on social media websites in order to make them a more conscientious zone for the world's youth. Many of the young internet users are taking drastic measures for reasons that could have been avoided. The application would flag the bully comment in real-time and notify the poster.

Proposed Solution

Our proposed solution is a classification prototype that predicts if a user inputted string is a bully comment or not-questionable in real-time. The analyzer is an ensemble model that is fed the user input as test data. This ensemble consists of training three models. These are Multinomial Naïve Bayes, Random Forest, and Stochastic Gradient Descent (SGD). The input is tested on each of these models and a majority vote is taken. The outcome of the prediction is displayed as a label on the GUI in real-time as the text is typed by the user.

The dataset is split 80/20 for training and testing of the models. The messages are tokenized, lowercased, removed of stopwords, and stemming is carried out. The bag of words matrix is created, and is normalized across documents with inverse document frequency (TFIDF). Once the training of the three models is complete and the ensemble is created, they are evaluated. Each model, including the ensemble, is 10-fold cross validated for accuracy and a confusion matrix is created for calculating the precision, false positive rate, recall. ROC curves are also drawn for each model.

Dataset and Features

The dataset used is from Kaggle.com and concerns a single class problem with regard to classification. The dataset consists of 3494 social media comments evenly split between bully comments and not-questionable comments. The dataset was pre-labeled. Each comment has a corresponding label of 0 or 1, 0 indicating a not-questionable comment and 1 indicating a bully comment. We processed the text from the comment attribute and the corresponding label to train our classification models. Since the data came pre-labeled by Kaggle, there was some noise within the training data labels.

Process Flow / Architecture

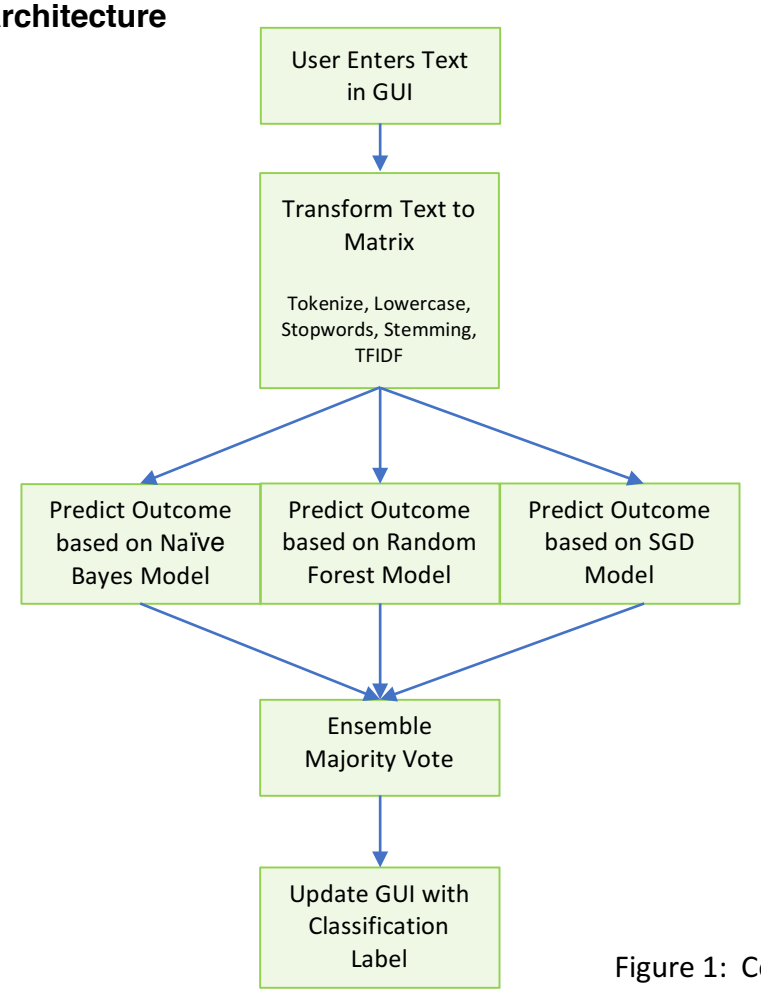

Figure 1: Comment Flagger Process

When the user enters text into the GUI textbox, the string must be converted into the same vector-space as the models that were trained on the features of the dataset. So, the string is preprocessed in the same manner as the training data with tokenization, lowercasing, stopword removal, stemming, and term frequency multiplied

by TFIDF. Now that the user string has been transformed, it is tested by the ensemble model to make a prediction of the user-entered string's class label and display it to the user in real-time.

Result and Evaluation

This implementation is a prototype to prove that user entered comments can be flagged as "Not-questionable" or "Bully" with better accuracy than baseline, which is 50%. This is a relatively simple model that contains a bag of words model and TFIDF as document features to train the models.

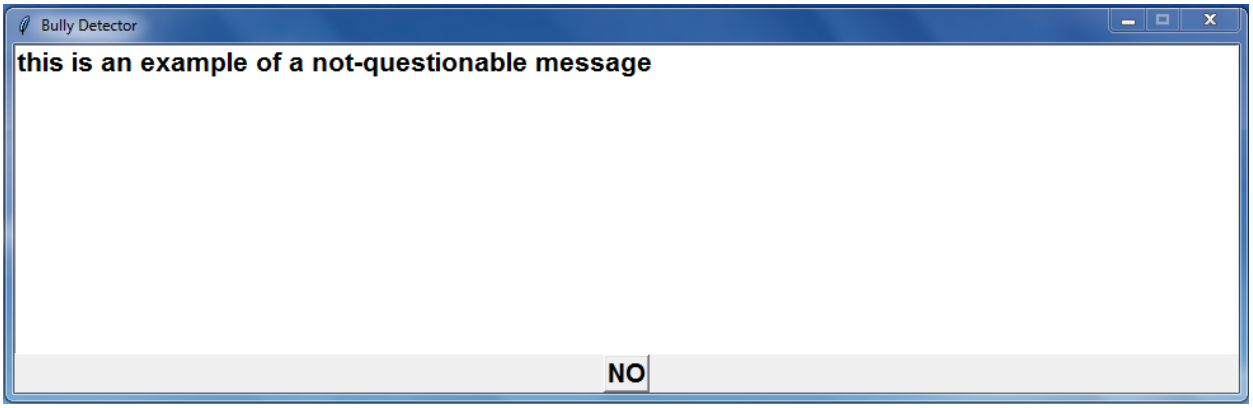

Figure 2: Example not-questionable message

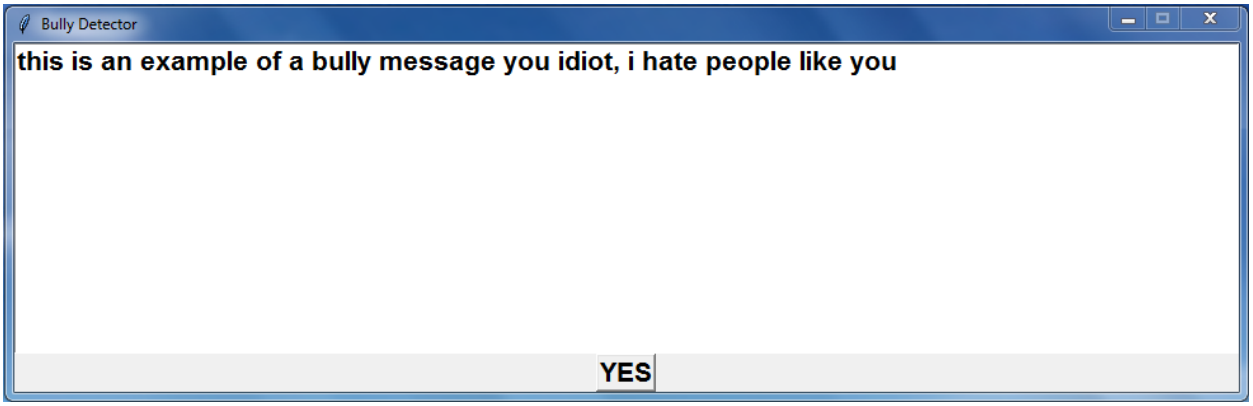

Figure 3: Example bully message

We evaluated our models and ensemble by calculating the accuracy, precision, false positive rate, recall and ROC curve for each model. These evaluation results are shown below. The model with the highest accuracy is the ensemble of the three models. The false positive rate (FPR) is the rate of flagging a not-questionable comment as a bully comment. This is an important metric for our system because we do not want to alter a non-bully user's experience on social media sites. The ensemble's FPR is much

lower than Naïve Bayes, while Naïve Bayes has much higher recall, which is the percentage of the total bully comments that were flagged as bully by the analyzer. The ensemble method achieves the highest area under the ROC curve (shared by SGD and Random Forest).

	Accuracy	Precision	FPR	Recall
Naive Bayes	76.25%	70.38%	32.14%	82.99%
SGD	77.73%	81.98%	14.01%	69.25%
Random Forest	76.99%	82.39%	13.74%	69.85%
Ensemble	78.22%	80.33%	16.21%	71.94%

Figure 4: Evaluation Metrics Per Model

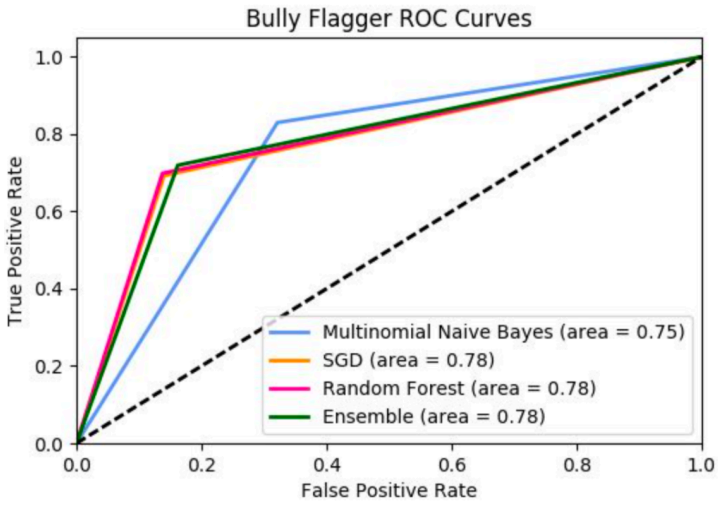

Figure 5: ROC Curves Per Model

Conclusion

As the number of users on social media sites continues to grow, many nefarious users will be tempted to act as cyber bullies. This prototype implementation successfully identifies bully comments as they are entered by the user. The results of the prototype is a comment analyzer that performs with 78.22% accuracy (28.22% above baseline), and a FPR of 16.21%. The GUI responds to the user's text inputs and updates the classification label in real-time.

Some limitations of the project are the small amount of data used and the subjectivity of the dataset's classification labels. Having a larger training dataset as well

as a separate testing dataset would allow for training stronger models while avoiding overfitting. Additionally, reducing the noise within the datasets would help to improve the evaluation metrics of the models. Finer tuning for the individual models used within the ensemble would be investigated as well as taking n-grams into account while pre-processing the data, as these may improve the metrics as well.

https://github.com/aarondevlin/BullyCommentFlagger.git

References

Maite Taboada, Julian Brooke, Milan Tofiloski, Kimberly Voll, Manfred Stede. Lexicon-Based Methods for Sentiment Analysis. Association for Computational Linguistics, 2011.

Jun-Ming Xu, Xiaojin Zhu, Amy Bellmore. Fast learning for sentiment analysis on bullying. Association for Computing Machinery, 2012.

Thomas Hofmann. Learning the Similarity of Documents: An Information- Geometric Approach to Document Retrieval and Categorization. MIT Press, 2000.

Python Libraries: Tkinter, Sklearn, Numpy, Pandas