Detecting Traces of Bullying in Twitter Posts Using Machine Learning

Caroline Jin, Harpreet Kaur, Amena Khatun, and Sitara Uppalapati
MIT Beaver Works Summer Institute

Abstract

According to research conducted by the University of Wisconsin, there are over 100,000 bullying-related tweets sent through Twitter every week (Xu, Jun, Zhu, & Bellmore, 2012). These social media posts show the prevalence of bullying both online and in the physical world. Researchers find that victims of bullying are more likely to have symptoms of depression (Hodges & Perry, 1999). Oftentimes, these tweets are unreported, leaving the victim to face mental consequences when kept in a bullying situation. Natural language processing (NLP) and machine learning, however, can efficiently detect bullying-related tweets and bring help to those in need.

The purpose of this research was to explore various methods of identifying bullying related tweets posted by various groups, including perpetrators and victims, through NLP and machine learning. We produced word embeddings (word vectors) and used the term frequency-inverse document frequency (TF-IDF) algorithm. Then, we trained various machine learning models, including multinomial Naive Bayes classifier, convolutional neural networks (CNN) and recurrent neural networks (RNN), to improve the predictions for the labels of tweets and observe patterns among them. For binary classification (bullying traces vs. no bullying traces), our CNN model produced the highest area under receiver operating curve (AUROC) of 0.924. AUROC was a more comprehensive metric which accounted for our skewed dataset. For multiclass classification, the tweets were separated into six classes: NA (the tweet was not bullying-related and was not annotated), self-disclosure, report, accusation, denial, and cyberbullying. Our RNN model produced AUROC value for each class, ranging from .854 to .936, for multiclass classification.

We aim to implement a script to Twitter to help identify users who are being impacted by bullying and may require mental health assistance. This research has important real world applications because as social media usage increases so do the number of malicious posts online.

Introduction

Depression is a major issue among youth mainly due to bullying. High school and middle school students surveyed by the Cyberbullying Research Center report having higher rates of self-deprecating and suicidal thoughts after experiencing bullying (Hinduja & Patchin, 2018). Bullying emerges from the negative emotions that rise from relationship problems including breakups, intolerance, and ganging up on one individual. These problems result in teens and adolescents experiencing anxiety, fear, depression, and low self esteem in their everyday lives.

One key factor contributing to the expansion of bullying and thus depression is social media. As technology and social media infiltrate our lives, more than 1/3 of the youth become victims of bullying (*Cyberbullying Statistics*, n.d.). 71% of young people say that they are concerned about cyberbullying (*Cyberbullying Statistics*, n.d.). Furthermore, through research conducted by the University of Wisconsin in 2011, 250 million public tweets are sent daily during the study—a number almost 10 times the population of the state of Texas (Xu et al., 2012). This problem must be tackled in order to prevent further cases of bullying.

Previously conducted research related to our topic gives us some insight on how to proceed. The research, "Using Machine Learning to Detect Cyberbullying," focused on Formspring.me, a question-and-answer website, and implemented a decision tree and instance based learner model to determine cyberbullying content (Reynolds, Kontostathis, & Edwards, 2011). Researchers have also worked on detecting bullying content in gaming chat rooms like the World of Tanks. Using SQL database queries and AI-based sentiment text analysis services, they found the prevalence of offensive language and racist sentiment (Murnion, Buchanan, Smales, & Russell, 2018).

Few research papers in machine learning have focused on bullying in both the real and virtual world. Furthermore, few research has considered the specifics of bullying, particularly on the different roles involved in a bullying situation. Researchers from the University of Wisconsin considered bullying in both worlds and these complex relationships while analyzing tweets containing bullying traces. They utilized a simple Naive Bayes classifier and a support vector machine (SVM) to classify whether a tweet was bullying-related and what kind of response the tweet was (Xu et al., 2012). The research was mostly a broad overview of different NLP techniques and machine learning models that could be used in studying bullying. Our research focuses specifically on which models are more effective in examining various tweets and detecting traces of bullying among them.

Materials

We utilized a labeled public dataset from the University of Wisconsin. The dataset had 7,322 data points, each one containing the tweet ID, user ID, and labels including 'Bullying Traces?,' 'Type,' 'Form,' 'Teasing?,' 'Author Role,' and 'Emotion.' The labels we focused on in this research were 'Bullying Traces?' for our binary classification and 'Type' for our multiclass classification. 'Bullying Traces?' indicated whether or not a tweet contained content related to bullying and used 'n' or 'y' as its values. In the 'Type' column, tweets were labeled as either 'NA' (the tweet is not a bullying trace and was not annotated), 'self-disclosure,' 'report,' 'accusation,' 'denial,' or 'cyberbullying.'

We extracted the raw tweets using the provided tweet IDs. The first

	Unnamed: 0	Tweet ID	User ID	Bullying_Traces?	Type	Form	Teasing?	Author_Role	Emotion	Text
0	0	107688644067856384	185389094	1	1	1	1	1	1	@bellathorne143 i herd that you got bullied wh...
1	1	102206417217392640	226320672	0	0	0	0	0	0	Bullying: O gesto mais idiota, estpido e irrac...
2	2	102779484725448704	297557032	0	0	0	0	0	0	The Bully at School Goes High Tech Part 1: \n...
3	3	108676972149874688	157724561	0	0	0	0	0	0	Esse @Felipemath fazendo bullying comigo! Haha'
4	4	106590106873372672	62179998	0	0	0	0	0	0	AUISHUAHS eu e o @wallace_mancha tiramos o dia...
5	5	109034091743154176	177913822	0	0	0	0	0	0	@Looweihao @haoyangg @sleepybed Hello, since w...
6	6	102533497637437441	70412906	1	1	1	1	1	3	For those keeping score at home- cousin who ju...
7	7	1036790088556912 64	11363462	0	0	0	0	0	0	BETTER ANTI-BULLYING AD SLOGANS: Hey, Bullies....
8	8	105202476927549441	293593896	0	0	0	0	0	0	@luttylutz bahh lo kaga demen dia juga? Gue ki...
9	9	107278521805713408	150964152	0	0	0	0	0	0	cara3: bully iam gara2 uda di ucpin tpi gk bls'
10	10	107478741424414720	360086723	0	0	0	0	0	0	http://t.co/nbBZDxf Please help to stop workpl...
11	11	103471384973418497	219025244	0	0	0	0	0	0	@lucy_robbo yes on thur, she said she might, w...
12	12	106840582453870592	244569315	0	0	0	0	0	0	@bbuk @brianofficial bobby needs threw out tha...
13	13	101324602789208064	210803457	1	1	1	1	1	3	don't you get it? This is me getting rid of yo...
14	14	105799405784936448	264606687	0	0	0	0	0	0	No bullying se eu jogar um tijolo na sua cara...

Figure 1. Sample of Filtered Bullying Dataset

part was checking if the tweet existed; we did this by checking if the url (https://twitter.com/statuses/TweetID) for each tweet was valid. The second part involved removing tweets that were private using Twitter Developer Platform through tweepy. Through this process, 5,443 tweets were left in the first round, and 3,755 tweets remained in the second round. Figure 1 shows a sample of the dataset we obtained after the second round. Using the remaining tweets, we applied NLP techniques to extract specific features and trained various machine learning models.

Methods

With the tweets, we formatted the data into both binary and multiclass by changing the textual labels to numerical representations. Afterwards, we ran various natural language processing algorithms such as TF-IDF, stop words, and word-vectors. We used a multinomial Naive Bayes, recurrent neural network, and convolutional neural network to determine which model(s) would best identify the tweets that hint toward bullying.

Multinomial Naive Bayes with TF-IDF and Stop words

TF-IDF was a simple NLP algorithm we used to process our text. This algorithm is similar to bag-of-words, which considers only the frequency of each word in the tweets. TF-IDF extends this idea by also considering the number of tweets that contains the word. In doing so, TF-IDF weighs words like 'the' and 'a' less heavily than words that better differentiate the bullying-related tweets from the non-bullying-related tweets.

We applied this algorithm to our data and inputted the processed data into a multinomial Naive Bayes classifier. We also considered removing these common words, also known as stop words, before using TF-IDF algorithm. We took out the stop words from tweets but kept words such as 'I' and 'you' as a point of view could also be useful for classification. We then fed the cleaned tweets into TF-IDF and a multinomial Naive Bayes classifier.

Word Embedding: Word2Vec

Word embeddings map each word to its corresponding vector. We used Word2Vec, which took raw text as input and learned a word by considering its surrounding context

or predicted a word given its surrounding context using gradient descent with randomly initialized vectors. Word2Vec used different vectors for word embeddings depending on whether it was the word we conditioned on or the word we tried to predict. The probability we maximized then was:

$$P(V_{out}|V_{in}), \text{ where } V_{out} \text{ is the output word and } V_{in} \text{ the input.}$$

The interesting property of word vectors obtained from this equation was that they encoded not only syntactic but also semantic relationships between words. Not only were similar words close to each other in the vector space (as measured by some norm), but word analogies were reflected by the difference between word vectors. This property known as 'additive compositionality' referred to the linear structure in the vector space that allows analogical reasoning. Word vectors thus can be seen as representing the distribution of the context in which a word appears, and the sum of vectors roughly represents an AND concatenation.

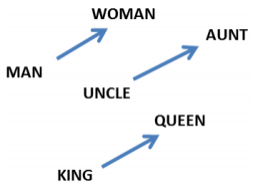


Figure 2. Example vectors with corresponding words

For our research, we used Stanford's pretrained global vectors for word representation dictionary, which held 400,000 words with their corresponding vectors of length 100 (Pennington, Socher, & Manning, 2014). Before converting the tweets into vectors, we made the words all lowercase as the dictionary contained only lowercase words. Furthermore, we removed words that were not in the dictionary; these words were often long strings of repeated letters like 'hahahahaha,' references to other users like '@bellathorne,' or html links. After preprocessing, we converted the words in the tweets into word vectors using the Stanford dictionary. We stored each tweet into a vector of size 30 by 100 (30 being the max number of words per tweet and 100 being the length of each word vector). For tweets with less than 30 words, zero vectors were placed. The resulting vector for all of our data was of size 3755 by 30 by 100.

Neural Networks

We divided the Word2Vec data into training, validation, and test of 2403, 601, and 751 tweets respectively. Before applying our data into our models, we transformed our labels from strings to numerical values. For our binary class, we relabeled 'n' and 'y' as 0 and 1 respectively. The multiclass labels changed from 'NA,' 'self-disclosure,' 'report,' 'accusation,' 'denial,' and 'cyberbullying' to 1-D vectors of length 6 using one-hot encoding. Using this transformed data, we developed two different neural network models for each of our binary and multiclass classification.

For binary, we used a RNN model containing a long-short term memory (LSTM) layer of 100 perceptrons and a fully-connected layer. LSTM can improve the classification

of sequential data like text as these units enable the model at arbitrary points of time to remember the order of the sequence. The second model was a CNN of four layers. CNN models are often used in image processing but also can be used in NLP. Our model contained 2-D convolutional kernels, 2-D max pooling kernel, a flattening layer, and a fully-connected layer. This model is shown in Figure 3.

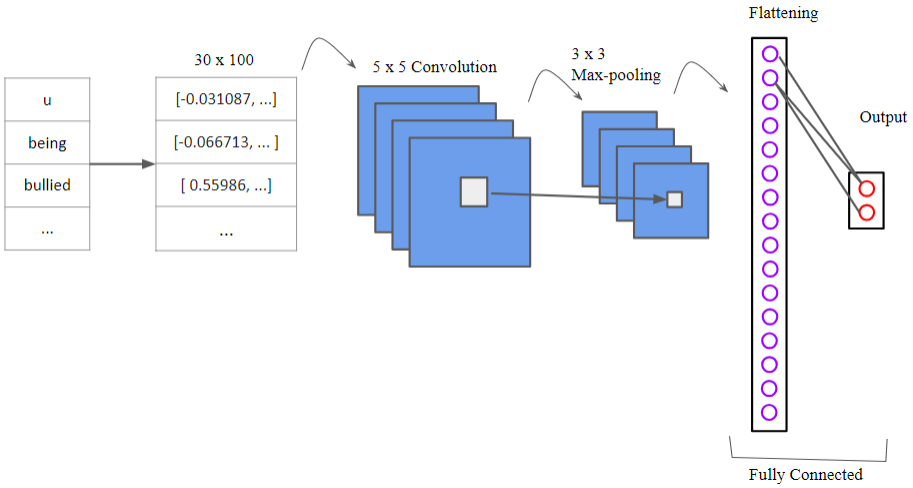


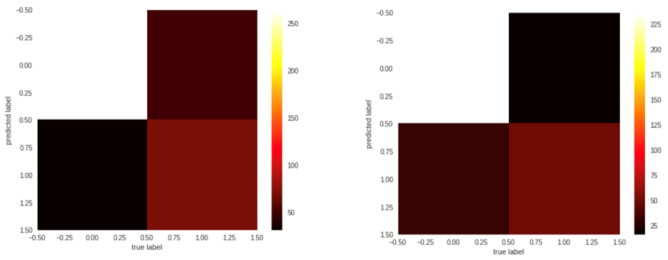
Figure 3. CNN Model used for binary classification

We used similar models for multiclass changing just the input and output shape to our desired dimensions. For each model, we trained for 40 epochs with a batch size of 64. After 40 epochs, the learning curve of the model followed an asymptotic behavior, so 40 epochs was a reasonable period to train our models.

Results

Since our task was identifying potential tweets with bullying traces, a classifier that would classify actual bullying tweets as non-bullying would not be optimal. Thus, we considered the precision rate through the confusion matrix and the area under the curve (AUC) as our evaluation metrics. For our binary classification, our RNN performed slightly worse than the multinomial Naive Bayes and CNN based on a lower AUC of .772. Figure 4 shows similarities between binary classification of the multinomial Naive Bayes and CNN based on our confusion matrix. From top to bottom and left to right, the confusion matrix referred to the number of cases that were true positives, false positives, false negatives, and true negatives. In both cases, there were 606 tweets that were non-bullying traces and 147 tweets that were bullying traces. This disparity in the number of tweets contributed to the shade representing the true negative being much darker than the shade representing the true positive in both confusion matrices.

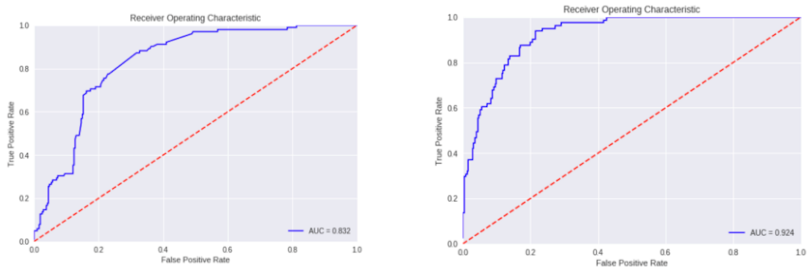
The multinomial Naive Bayes classifier tended to have a higher number of cases that were false positives rather than false negatives whereas the CNN classifier tended to have a higher number of false negatives rather than false positives. For our classification task, we wanted to capture all potential bullying-related tweets rather than miss tweets that were



(a) Multinomial Naive Bayes with TF-IDF (b) CNN

Figure 4. Color Map of Confusion Matrix for Binary Classification. Main diagonal from top to bottom refers to the number of true positives and true negatives. Lighter color means more data is being classified in that class

actually bullying-related. In this aspect, the CNN was the better choice. In Figure 5, the receiver operator curve (ROC) curve showed that the Multiple Naive Bayes classifier had an AUC of .832 while the CNN had had an AUC .924. This metric further revealed our CNN model as the better classifier for binary classification.



(a) Multinomial Naive Bayes with TF-IDF (b) CNN

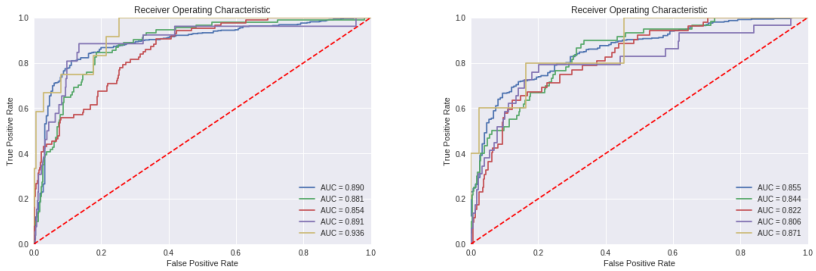
Figure 5. ROC for different binary classifiers

For the multiclass, the bullying traces were separated into six classes: NA, self-disclosure, report, accusation, denial, and cyberbullying. Our dataset had a lack of tweets that indicated cyberbullying. Out of the 3755 tweets, only eight were labeled as 'cyberbullying' with four tweets in the training set, two in the validation set, and two in the test set. Due to the lack of data, neither multiclass models could classify the few cyberbullying cases. Thus, for our comparison, we focused on the remaining five classes.

Unlike our binary classification, our RNN classifier tended to perform better in the five classes as shown in Figure 6. The AUROC for RNN was between .854 and .936 for all the classes whereas the AUROC for the CNN was between was .806 and .871. Each class for the RNN had a higher AUROC than that for the CNN, showing that our RNN model was a stronger classifier for this task.

Conclusion

To demonstrate how our models worked, we created a dummy twitter account `Medlytics_5Test`. The account was followed by us. We then posted some text that we



(a) RNN (b) CNN

Figure 6. ROC Curve for each of the multiclass classifiers. The key from top to bottom, the color corresponds to NA, self-disclosure, report, accusation, and denial respectively. *Note: cyberbullying ROC curve was not considered due to lack of data.*

thought had traces of bullying and some that did not. Every post was extracted by our model using tweepy. The model would get a timeline of the user's posts and pick out the most recent post. Then, the text was formatted based on the methods described in *Methods* section. Afterward, the models that gave the highest accuracy was utilized to get the probability of the text containing bullying traces. The model would report the probability using the follower account.

The model did not predict that a post was bullying or not because in order to do that it would have to understand the context in which the post was written. However, by reporting the probability of it being a bullying post it drew attention the the post so other users could read it and determine if it counted as bullying or not and took the appropriate action.

Discussion

The problems faced while attempting to address cyberbullying on Twitter utilizing machine learning were multifaceted. Because the labeled data was from 2012, many of the posts were either deleted or made private. In the end, the data frame had about 4,000 data points which was a small sample size considering the number of interactions that took place on Twitter. Additionally, the Stanford dictionary was limited to common words and included only text emoticons, such as ':)' or ':p.' Many of the tweets had keyboard emojis and usernames that were hard to vectorize. The data set had few cases involving cyberbullying and was also biased because the labeling was very subjective to the researchers who had preprocessed the data. Lastly, there were contextual problems because it was difficult to identify a tweet as bullying without understanding the context of the situation.

The Stanford dictionary contained words from multiple languages including Spanish, meaning our model was able to detect bullying present in tweets of different parts of the world. But as mentioned before, the contextual problem was exacerbated when addressing different languages.

To take this project to the next level, more tweets needed to be extracted from Twitter, analyzed, and labeled as containing traces of bullying or not. A dictionary with the ability to turn abbreviations, emojis, and slang words to vectors, a way to account for spelling mistakes and contextualizing our data, and greater computing power would improve the

functionality of the model. Lastly, adapting the algorithm to Twitter would allow us a chance to see how the algorithm can help with bullying.

References

Cyberbullying statistics. (n.d.). Retrieved from `https://enough.org/stats_cyberbullying`

Hinduja, S., & Patchin, J. W. (2018, Aug). Connecting adolescent suicide to the severity of bullying and cyberbullying. *Journal of School Violence*. doi: 10.1080/15388220.2018.1492417

Hodges, E. V. E., & Perry, D. G. (1999, Apr). Personal and interpersonal antecedents and consequences of victimization by peers. *Journal of Personality and Social Psychology*, *76*(4), 677âĂŞ685. doi: 10.1037//0022-3514.76.4.677

Murnion, S., Buchanan, W. J., Smales, A., & Russell, G. (2018). Machine learning and semantic analysis of in-game chat for cyberbullying. *Computers Security*, *76*, 197 - 213. Retrieved from `http://www.sciencedirect.com/science/article/pii/S0167404818301597` doi: 10.1016/j.cose.2018.02.016

Pennington, J., Socher, R., & Manning, C. (2014). Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (emnlp)* (pp. 1532–1543).

Reynolds, K., Kontostathis, A., & Edwards, L. (2011, Dec). Using machine learning to detect cyberbullying. In *2011 10th international conference on machine learning and applications and workshops* (Vol. 2, p. 241-244). doi: 10.1109/ICMLA.2011.152

Xu, J.-M., Jun, K.-S., Zhu, X., & Bellmore, A. (2012). Learning from bullying traces in social media. In *Proceedings of the 2012 conference of the north american chapter of the association for computational linguistics: Human language technologies* (pp. 656–666). Stroudsburg, PA, USA: Association for Computational Linguistics. Retrieved from `http://dl.acm.org/citation.cfm?id=2382029.2382139`