Ben Prescott
Second Research/Programming Assignment
MSDS 453 2021SP
4/27/2021

Reddit Utterance Preparation for

Identifying Cases of Cyber Bullying

Abstract

In this paper I explore a reduced version of the Reddit corpus, consisting of 119,889 'speakers' (user information), 8,286 'conversations' (post information), and 297,132 'utterances' (comment information). My focus is on the corpus' utterances, which provides individual speaker comments that can be used for future research into sentiment analysis. I explore three different methods to generate document vectors: 'analyst judgement', term frequency-inverse document frequency, and Doc2Vec, with a goal of being able to identify which subreddit a comment originated from.

Three Random Forest Classifiers, using a Count vectorizer, TF-IDF vectorizer, and Doc2Vec, are then fit to determine which method of vectorization provides the best performance. While the corpus does not have categories, I used each documents' individual subreddit as the target category.

Introduction

According to survey results published by Statista in 2019, 36.5% of middle and high school students (ages 12 to 17) reported having been bullied online. In fact, cyber bullying is considered a form of defamation and harassment. As of the end of 2018, 48 U.S. states were recorded having laws that include cyber bullying or online harassment.

Cyber bullying is a social issue that introduced risk of long-term psychological problems. While cyber bullying can affect a person of any age, the state of technological advancement places younger age groups at greater long-term risk. This poses a challenge for parents as well as

opens the discussion of overarching internet governance and finding ways to combat cases of cyber bullying.

This research is one very small step in a larger solution around attempting to identify cases of cyber bullying, specific to the website/app Reddit. The results of this research are initial models that can assist in identifying correlation between utterance structure and specific subreddits.

Literature Review

Others have explored similar research using data from Twitter or chat websites, all of which required either manually labeling the data as bullying / not bullying, or manually assigning weights/scores to data based on found negative content. Having manually labeled the data, the different approaches were able to leverage supervised learning algorithms to successfully train a model, with one showing a 91% true positive rate with a Random Forest Classifier (Talpur & O'Sullivan, 2020). One of the issues with these approaches, as is with any manual labeling effort, is the risk of introducing a subjective bias.

A few approaches leveraged pre-trained models, such as NLTK's VADER (Valence Aware Dictionary and sEntiment Reasoner), as well as unsupervised methods. My future research will look to compare both a pre-train VADER model and unsupervised methods.

Methods

This research started with a corpus named 'reddit-corpus-small' available from the ConvoKit Python library and focuses on the available utterance information (individual comment metadata). The corpus contains 297,132 utterances across 100 of the most active subreddits, collected for only the month of September 2018.

I started by extracting each utterance's ID, the direct URL to the utterance, the Speaker (user) who posted the comment, the Body (text) of the comment, and the Subreddit in which the comment was posted. The data then went through processing to remove Reddit's automated bot's (AutoModerator) comments from the corpus, as well as any empty, deleted, or removed comments. Each utterance's Body text was then manipulated to remove punctuation, remove stop words, convert all characters to lowercase, identify only words that were alphabetical, and only keep words longer than three characters long.

The cleaned corpus was then saved to a JSON lines file and added to a Pandas DataFrame to provide ease of searching and visualization. The top 20 subreddits with the most utterances were then used for the remainder of the research, with the Technology subreddit being used for vector comparison, and the Technology and TodayILearned subreddits for training the Random Forest Classifiers.

To determine the 'Analyst Judgement' vectors, I combined all documents from the Technology subreddit and chose four of the most common terms that I felt would be the most prevalent – people, companies, government, and money. Each of these four terms were then identified and counted in the first 20 documents. The Term Frequency – Inverse Document Frequency approach required combining individual tokens back into a comma-delimited list of strings. Sci-kit Learn's TfidfVectorizer was used to generate the TF-IDF vectors, matched against the four identified words for the same 20 documents, and added to a DataFrame. Doc2Vec was then used for the third approach, generating vectors for the same 20 documents. While Doc2Vec does not retain the associated terms, the vector size generated was kept to four to retain a similar shape to the earlier two approaches.

Three Random Forest Classifiers were also trained using utterances from the Technology and TodayILearned subreddits. The utterance data was not previously categorized, so the subreddit the utterance was assigned to was used as the category/label. To generate the train/test tokens and text the corpus of utterances was shuffled, then a random selection of 500 documents' Body information was retrieved for each train and test. Each utterance's associated subreddit was saved as the label. Using Scikit-Learn's RandomForestClassifier, a TF-IDF, Counter Vectorizer, and Doc2Vec classifier were all trained, and performance measured using their F1 score.

Results & Conclusions

The models were tested using vector sizes of 50, 150 and 200. The results of the three different classifiers show that the best overall algorithm is Doc2Vec, with an F1 score anywhere from 5% to 11% more accurate than Count and TF-IDF vectorizers. Using the three vector sizes as a guide, the Doc2Vec F1 score continued to increase as the vector size increased, where-as the Count vectorizer and TF-IDF vectorizer both peaked at 150 vectors and saw a degradation in performance at 200 vectors. I included a fourth vector size of 400 to show that Doc2Vec continues to improve, as the others show negligible changes. However, it is worth noting that the Doc2Vec approach is stochastic in nature and results may vary.

Further refinement of the classifier' hyperparameters is needed, but the initial results are promising for continued exploration of Doc2Vec. This research provides a means of leveraging different algorithms to identify which subreddit an utterance may belong to, which will aid in research into cyber bullying on Reddit. Additional research will be focused on sentiment analysis approaches leveraging the utterance data, with a goal of identifying utterances that may be a cyber bullying risk, as well as which subreddit the utterance is likely to belong to.

References

Brownlee, J. 2019. Deep Learning for Natural Language Processing: Develop Deep Learning Models for Natural Language in Python. Section I Introductions (pages iv–ix) and Section II Foundations (pages 1–33).

Gensim: Topic modelling for humans. (2021, April 29). Retrieved May 02, 2021, from https://radimrehurek.com/gensim/auto_examples/tutorials/run_doc2vec_lee.html

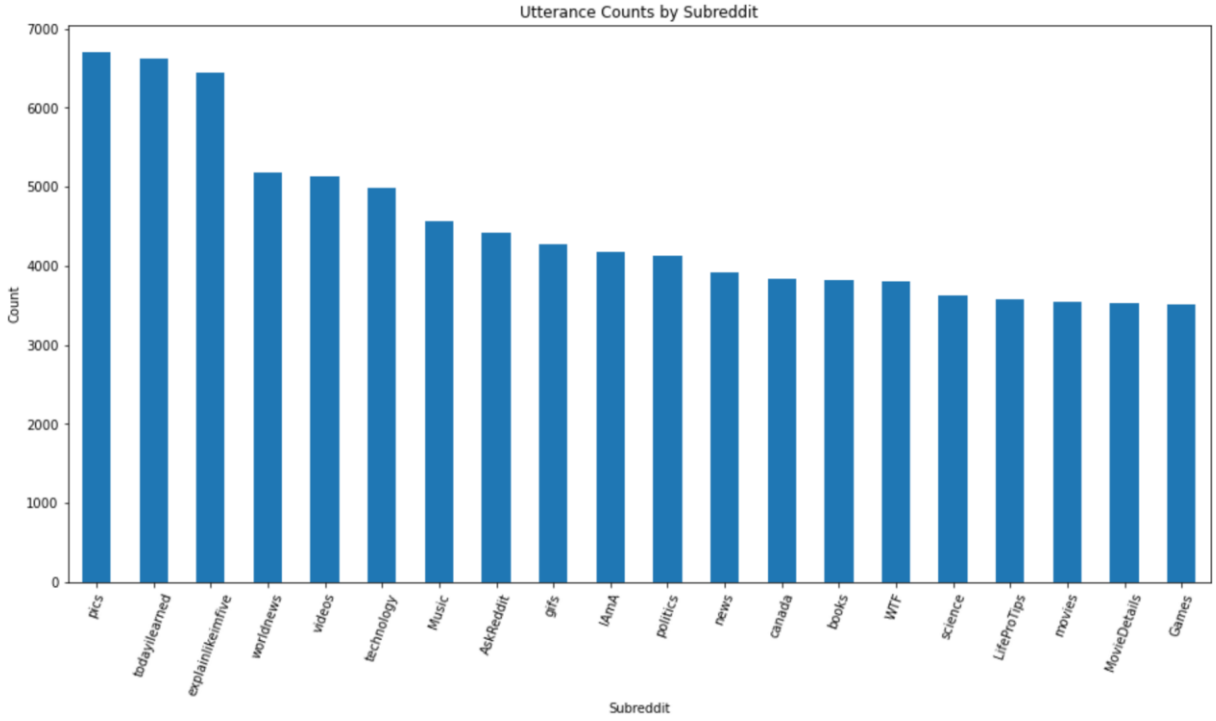
Johnson, J. (2021, January 25). U.S. States with state cyber bullying laws 2018. Retrieved April 27, 2021, from https://www.statista.com/statistics/291082/us-states-with-state-cyber-bullying-laws-policy/

Johnson, J. (2021, January 25). U.S. student cyber bullying victimization RATE 2019. Retrieved April 27, 2021, from https://www.statista.com/statistics/509327/student-cyber-bullying-victimization-rate-usa/

Talpur, B. A., & O'Sullivan, D. (2020). Cyberbullying severity detection: A machine learning approach. *PLOS ONE, 15*(10). doi:10.1371/journal.pone.0240924

Appendix

Utterance Counts by Subreddit



'Analyst Judgement', TF-IDF, Doc2Vec Vectorization respectively (First 20 Docs)

	id	people	companies	government	money
0	e57wo99	1	0	0	0
1	e57yqwn	1	0	0	0
2	e57zyvr	1	0	0	0
3	e584g9n	0	0	0	1
4	e5888f2	4	0	0	3
5	e588iks	5	0	0	0
6	e58995l	5	1	0	0
7	e589caj	0	0	0	1
8	e58a1e7	4	1	0	0
9	e58al3o	2	0	0	1
10	e58aohd	1	0	0	0
11	e58bcgd	3	0	0	0
12	e58be19	6	2	0	1
13	e58bm4c	1	0	0	0
14	e58bv4m	3	0	0	0
15	e58c3iy	0	1	0	0
16	e58dcse	1	0	0	2
17	e58dm6f	0	1	0	1
18	e58dtds	0	0	0	1
19	e58er4h	0	1	0	0

	id	people	companies	government	money
0	e57wo99	0.142247	0.000000	0.0	0.000000
1	e57yqwn	0.162799	0.000000	0.0	0.000000
2	e57zyvr	0.112136	0.000000	0.0	0.000000
3	e58bm4c	0.193906	0.000000	0.0	0.000000
4	e584g9n	0.000000	0.000000	0.0	0.100459
5	e5888f2	0.200862	0.000000	0.0	0.228899
6	e588iks	0.240686	0.000000	0.0	0.000000
7	e58995l	0.167349	0.045117	0.0	0.000000
8	e589caj	0.000000	0.000000	0.0	0.099602
9	e58a1e7	0.114352	0.038537	0.0	0.000000
10	e58al3o	0.080960	0.000000	0.0	0.061507
11	e58aohd	0.052267	0.000000	0.0	0.000000
12	e58bcgd	0.101020	0.000000	0.0	0.000000
13	e58be19	0.102674	0.046135	0.0	0.026001
14	e58bv4m	0.192370	0.000000	0.0	0.000000
15	e58c3iy	0.000000	0.220849	0.0	0.000000
16	e58dcse	0.022162	0.000000	0.0	0.067348
17	e58dm6f	0.000000	0.077062	0.0	0.086862
18	e58dtds	0.000000	0.000000	0.0	0.084331
19	e58er4h	0.000000	0.054674	0.0	0.000000

	id	0	1	2	3
0	e57wo99	0.020440	-0.127778	0.269114	0.328197
1	e57yqwn	0.203584	-0.137983	0.235256	0.148826
2	e57zyvr	0.107553	0.069734	0.118713	0.395098
3	e58bm4c	-0.147233	-0.246082	-0.086856	0.131670
4	e584g9n	0.629690	0.102222	0.571931	0.774755
5	e5888f2	0.710737	-0.008459	0.800173	0.697140
6	e588iks	0.662877	-0.014798	0.514976	0.997093
7	e58995l	0.530557	-0.365941	0.995237	1.803239
8	e589caj	0.586812	-0.135210	0.964407	0.819476
9	e58a1e7	0.612802	-0.166453	0.972527	1.964902
10	e58al3o	0.712747	-0.246366	1.319225	1.522101
11	e58aohd	0.801742	0.005346	1.012381	0.883957
12	e58bcgd	0.957569	-0.320321	1.763557	1.852302
13	e58be19	1.087078	-0.296398	1.482043	2.824023
14	e58bv4m	0.494766	0.083543	0.651050	0.679080
15	e58c3iy	0.257359	0.010762	0.259108	-0.052379
16	e58dcse	1.540648	-0.292114	1.961830	2.107255
17	e58dm6f	0.609635	-0.246925	0.645707	0.677244
18	e58dtds	0.409836	-0.157638	0.938132	0.828830
19	e58er4h	0.905316	0.056561	0.963303	1.229604

Classifier Performance F1 Scores

Classifier F1 Scores	Vector Size			
	50	150	200	400
Count Vectorizer	0.569	0.625	0.616	0.627
TF-IDF Vectorizer	0.572	0.638	0.586	0.612
Doc2Vec	0.673	0.678	0.687	0.691