Deeper Insight Into Tweet Classification for Bullying Traces. Also, Is It Possible To Make It More Memory Efficient?

Dario Mesić, Mladen Džida, Ana Petra Jukić

University of Zagreb, Faculty of Electrical Engineering and Computing
Unska 3, 10000 Zagreb, Croatia
dario.mesic@fer.hr, mladen.dzida@fer.hr, ana-petra.jukic@fer.hr,

Abstract

We analyzed the best approach to finding tweets that contain a bullying trace. During the research we found that BERTweet language model does the best job of creating representations of tweets for a task like classification and it achieved the best accuracy and recall on all models that we experimented on. To further understand why these BERTweet word embeddings gave such good results, we performed PCA analysis on the transformed sentence embeddings and found that already 86.28% of the variance lies in the first principal component. This discovery gave us an idea to improve memory requirements of these embeddings, where we tried classification of embeddings with ten times smaller feature space and achieved pretty much the same results. We believe this knowledge could substantially improve model training of dataset with large amount of tweets.

1. Introduction

Today, talking is replaced with chatting on the Internet, and friends are mostly created over social media accounts. We live each day by day, falling deeper in the hands of social media. This doesn't have to be so bad, but the problem occurs when some get emotional to the point where they hurt others over this virtual based program called social media. This is called cyber-bullying and topic of this article will be detecting and analyzing bullying traces which are social media posts which describe bullying experience and the posts we analyze will be taken from social media application called Twitter.

Figure 1 shows number of users who were bullied from Twitter through the period of nearly six months. As you can see, bullying traces had it's ups and downs but the curve always rose slightly upwards. This is due to the fact that people are joining each day and obviously cyber-bullying can't be eradicated forever. For all of these reasons, projects based on this topic can be useful for everyone.

Our task was to find as many ways as possible to detect harmful tweets and protect a person who is being bullied on social media from falling into depression, causing self-harm, becoming one of the bullies or even committing suicide. One research, that was made in 2007 by the National Crime Prevention Council and Harris Interactive Inc. found that 43 percent of the 824 middle school and high school-aged students surveyed in the United States had been cyberbullied in the past year(Aune, 2009). This was over a decade ago, when we experienced only the birth of social media and when Facebook wasn't even at it's peak. Imagine the results of this experiment done today.

Past cyber-bullying projects were focused more on achieving high accuracy scores on tweet test sets. For this project, we wanted to do something a bit different, we wanted to explore the existing methods for creating a sentence embedding from a tweet and find out which one would produce the best feature vector in a sense that it performs better on the task of generalization, where we experimented this classification task only on shallow models.

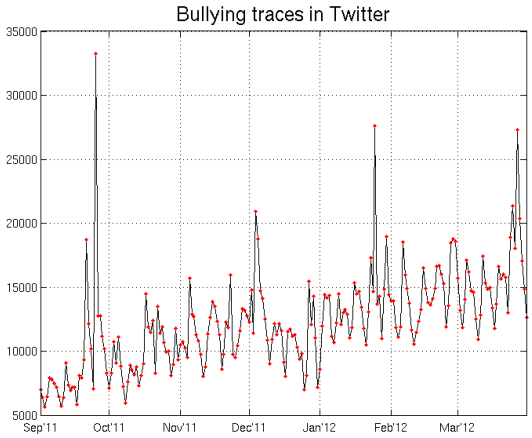

Figure 1: Bullying traces through six months.

Also, we used Principal Component Analysis to see where the most of the information from the data lies and by doing that we managed to discard less important features and achieve almost the same classification results with smaller feature vectors.

Related work will be presented in Section 2, followed by description of dataset and preprocessing that was used for our models. In Section 4 we will describe evaluation which consists of word and sentence embeddings, where we will explain further which one we used and why, and also our shallow models together with results. Section 5 will introduce PCA in which we used our embeddings and finally conclusion in Section 6.

2. Related Work

Cyber-bullying has been around for some while, so it doesn't really surprise us how many projects were already made on this subject. However, as we have already stated, many models that exist focused on accuracy and finding the best way to represent context and this can be tricky in many

situations where popular one is detecting sarcasm. An example of this kind of project is an article "Learning from Bullying Traces in Social Media" (Jun-Ming Xu, 2012). They extracted topics in bullying traces to facilitate understanding, but with that they only extracted words and placed them in a group based on sentiment, like "feelings", "verbal bullying", "school" and etc.

Project, that is similar to ours, based on word embeddings, is one made by Kostas Stoitsas and it's called "The use of word embeddings for cyberbullying detection in social media" (Stoitsas, 2018). Topic of this study is finding out how models based exclusively on word embeddings perform on the task of cyber-bullying detection compared to other techniques. Their experiments revealed that the higher the dimensionality of the vectors the better the performance of the model on cyberbullying detection is. However, the main difference between theirs and ours project is that they trained data on different models and used different evaluation metrics that consisted of word similarities.

3. Dataset

In this section we provide details about how were tweets gathered from Twitter and what each one of them had as a background information.

3.1. Data Collection

We used an existing dataset of tweets which consisted of tweet ids and then we used Twitter API to download these tweets. We gathered 7321 tweets, but we used 2522 tweets since only those were in English.

3.2. Preprocessing

Each tweet was normalized before transforming it into feature vector to have a consistent dataset. The process of normalization was run as follows: each user mention was replaced with @USER and all outer links were replaced with string "HTTPURL".

4. Evaluation

4.1. Word and Sentence Embeddings

We used sentence and word embeddings to represent the data. The different methods and language models that we used to create feature vectors were: BERTweet, Bag of Words(BoW), Doc2Vec, TF-IDF and InferSent language model. Specifically, for creating BERTweet sentence embeddings, we generated BERTweet word embeddings for each word in a sentence and computed average feature vector from these word vectors.

4.2. Models

After generating the sentence embeddings, they are used for detection of bullying traces in tweets which is a problem of binary classification. They were trained and evaluated on three different shallow machine learning models: logistic regression, XGBoost and KNN. For training these models we used 80% of the tweets and the rest we used for the test set.

Hyperparameters were not tuned, since our goal was not to raise the accuracy as much as possible, but rather to investigate the effect of different embeddings on the task of this binary classification. Deeper models were also not used for the same reason.

4.3. Results

Table 1 shows accuracy that different combinations of embeddings and models achieve on the test set. From the table we see that BERTweet and InferSent managed to get accuracies above 70%, where BERTweet performed better on logistic regression and InferSent performed better for XGBoost and KNN. Slightly worse results were achieved on Doc2Vec embeddings for all three models. TF-IDF and BoW performed rather poorly on all models, which we expected due to them being the simplest methods for extracting sentence embeddings. Logistic regression seems to be the most effective out of these three models for mostly all embedding types.

5. Using PCA To Gain Insights Into Tweet Representation and Classification

5.1. Analysis of Tweets With PCA

After the classification results, we performed PCA analysis on the computed feature vectors. With this analysis we got to see if tweets as points in this feature space are spread out in certain directions which preferably hold large amounts of variation in data. Table 2. shows the amount of explained variance for the first two principle components for each embedding type. BERTweet benefits the most from PCA, since it's first principle explains 86% of the variance. Meaning that BERTweet is useful to use if we want to have vectors of very small dimensions, but still keep most of the variance in the data. Reducing the feature vector will be explained in more detail in the next section. Figure 2. shows the reduction of BERTweet embeddings to three dimensions using the first three principal components. Green dots represent tweets that are considered bullying traces and the red dots are tweets where bullying wasn't detected. This figure shows that the BERTweet can separate these classes relatively well even when the embeddings are reduced to only three principle components. InferSent's first principal component explains 43% of variation in the data, which is also quite high. Suprisingly, BoW's first component explains 27% of variation, which is quite high and unexpected for this simple method of generating feature vector. TF-IDF's and Doc2Vec's first component explains less than 10% of variance. It is notable to mention high explained variance for BERTweet and InferSent feature vectors where they also gave the best classification results.

5.2. Classification of Tweets With Simpler Sentence Embeddings

We experimented with different number of principal components for each embedding type. This was done to experiment how much can we reduce the dimensionality of our embedding vectors, while still keeping as much of the variation in data as possible. Reducing the dimensionality with PCA can be useful for removing useless features and ensuring that features are less correlated. Greatest benefit for using PCA on word and sentence embeddings is reduced memory cost. Embeddings of words and sentences are usually large vectors (e.g. BERTweet produces vector

Table 1: Embeddings and f1 score on different models

Embeddings	Logistic Regression	XGBoost	KNN
BERTweet	0.71	0.66	0.58
TF-IDF	0.03	0.07	0.02
BoW	0.32	0.07	0.01
Doc2Vec	0.63	0.63	0.58
InferSent	0.69	0.70	0.72

Table 2: Amount of explained variance for first two principle components for different embeddings

Embeddings	Explained variance: first component	second component
BERTweet	0.863	0.015
TF-IDF	0.060	0.042
BoW	0.276	0.175
Doc2Vec	0.076	0.064
InferSent	0.436	0.124

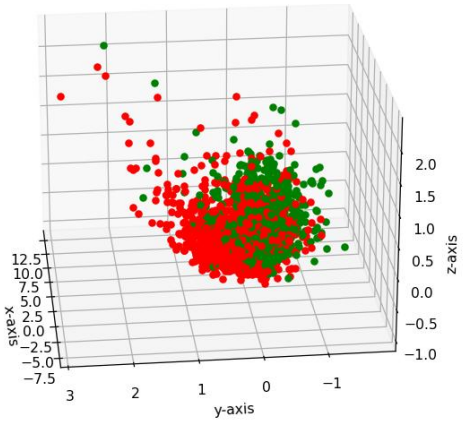

Figure 2: BERTweet embeddings reduced to three features.

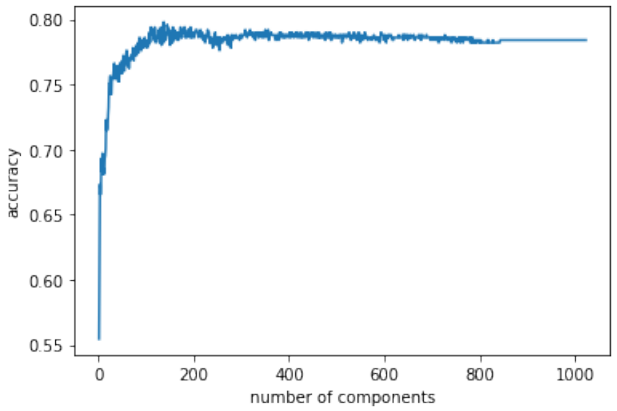

Figure 3: Accuracy levels on logistic regression for different number of components of BERTweet.

with 1024 features and InferSent produces vector with the length of 4096). With PCA we can reduce these huge embeddings to vectors less then 10 times their size while still keeping the same or even greater accuracy on models. Not only does this mean that the need for memory will be a lot smaller, but also that the embeddings can be trained and evaluated faster.

Specifically, For each of these embeddings, we generated simple embeddings with size ranging from 1 to their maximum length (number of data or number of original features). Then these simpler embedding were trained and evaluated on logistic regression to show how the different number of features affect the accuracy and f1 score of the model for tweet classification. Figure 3. shows accuracy levels corresponding to different size embedding for BERTweet. Figure 4. shows the same for f1 score. It is noticeable that at first, the larger the feature vector is the better accuracy and f1 scores are. However, that rise eventually stops and the graph stagnates. Optimum is found when using 137 features, where accuracy is 0.798 and f1 score is 0.766. This means that by reducing BERTweet vector from 1024 numbers to only 137 features we will improve the f1 score and the accuracy by almost 5%. Similar results were obtained with InferSent and Doc2Vec feature vectors, while TF-IDF and BoW didn't really behave this way and we couldn't make any conclusions for them.

6. Conclusion

In this paper, we explored different tweet representations for classifying bullying traces in tweets. We chose five popular word and sentence embedding approaches. Three simple machine learning models were trained and evaluated on the embedded data, where logistic regression showed the best performance. We then used principal component

Table 3: Embeddings and f1 score on different models

Embeddings	Logistic Regression	XGBoost	KNN
BERTweet	0.71	0.66	0.58
TF-IDF	0.03	0.07	0.02
BoW	0.32	0.07	0.01
Doc2Vec	0.63	0.63	0.58
InferSent	0.69	0.70	0.72

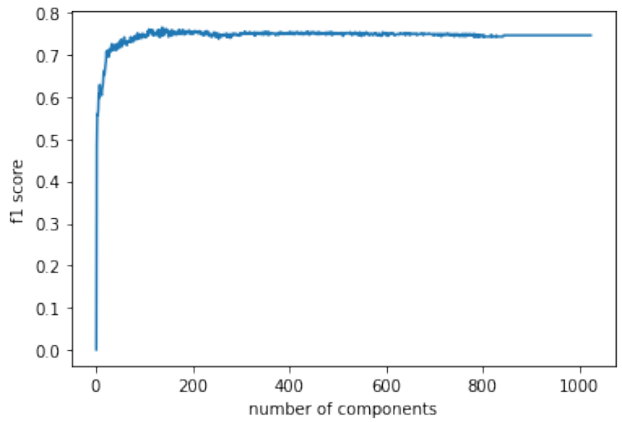

Figure 4: F1 scores on logistic regression for different number of components of BERTweet.

analysis to reduce dimension of embedded data. Results show that BERTweet and InferSent feature vectors can be transformed to low number of principal components which amount to high variation in data. Language model that especially caught our eye was BERTweet, which has 86.28% of data variance explained by the first principal component. Reducing these models to smaller dimensions is especially efficient for tasks where large amount of tweets need to be stored and trained. To confirm that reduction of embeddings to smaller dimensions does not decrease ability of generalization of models, we conducted classification of tweets with smaller embeddings on logistic regression. The results show that reducing sentence embeddings to appropriate size does not decrease accuracy or f1 score and sometimes even increases them.

While performing PCA analysis on TF-IDF and BoW feature vectors we noticed the potential for future work. Since features in TF-IDF and BoW vectors are numbers associated to specific words, principal components will be consisting of weights associated with each occurring word. With this is mind, there is a potential for interpretability in a way that we look for words with high weights in the first principal components and therefore we would get which words are important in achieving variability in the data.

References

Nicole M. Aune. 2009. Cyberbullying.

Xiaojin Zhu Jun-Ming Xu, Kwang-Sung Jun. 2012. Learning from bullying traces in social media.

Kostas Stoitsas. 2018. The use of word embeddings for cyberbullying detection in social media.