Cyberbullying Detection for Social Media

Rohan Mahajan, Adesh Gadge, Vinayak Kukreja, Anirudh Raman

IST 664 - Natural Language Processing

Prepared for Prof. Stephen Wallace

Completed on:
12/11/2019

Introduction and Overview:

Cyberbullying is a huge and rather sensitive topic in today's world. With social media at it's peak and the increasing number of users and engagement, cyberbullying has always been a topic that has come into the picture. Nowadays, the effects of cyberbullying are almost as worse as effects of physical crimes. Therefore, new laws and filtering on social media has been brought into the picture. Since the major demographic of social media now has children of ages 7-10 as well, social media must have a healthy and family-friendly platform.

For our analysis, we used Natural Language Processing Techniques to work on text data and combine them with unsupervised learning models to generate a cyberbullying detection system.

Our analysis had the following major steps:

- Data frame conversion
- Data Pre Processing
- Sentence Vectorization
- Exploratory Data Analysis
- Model Training, Testing and Evaluation
- Analysis and Model Comparison with TF - IDF Vectorizer

Dataset Description:

Our dataset consisted of 50 ids from Formspring.me that were collected in the summer of 2010. Formspring.me is a social networking website where users can ask and respond to questions. The content in the questions/answers remain unfiltered and therefore, it could range from shallow or funny to deep and insightful content. This also includes inappropriate content from users as the website remains unfiltered which has a potential of users getting cyberbullied. We therefore decided to use this dataset as it could be best used for detecting cyberbullying.

Our dataset consisted of 13159 entries and 2 columns, one being the content(sentence) variable which we will use to conduct our analysis on and the other being the label for whether the entry is an inappropriate entry (cyberbullying) or not. We already had the entries human-labeled which helped us for our classification models.

Data Preprocessing:

Data frame Conversion:

Our dataset was in XML format. We used an XML parser to extract our file and gather the data stored in it. We then converted it into a pandas data frame.

In [2]:
```
xtree = et.parse("XMLMergedFile.xml")
xroot = xtree.getroot()
```

In [3]:
```
c1=c2=0
text =[]
Bullying=[]
for child in xroot:
  for z in child:
     for x in z:

       if(x.tag=='TEXT'):
        flag= True
        c1=c1+1
        if x.text is not None:
          #print(x.text)
          text.append(re.search("Q: (.*)",x.text).group(1))
       if(flag):
        if(x.tag=='LABELDATA'):
          flag=False
          c2=c2+1
          for y in x:
            if(y.tag=='ANSWER'):
                Bullying.append(y.text)
```

In [6]:
```
df=pd.DataFrame({
    'Text': text,'Bullying':Bullying
})
```

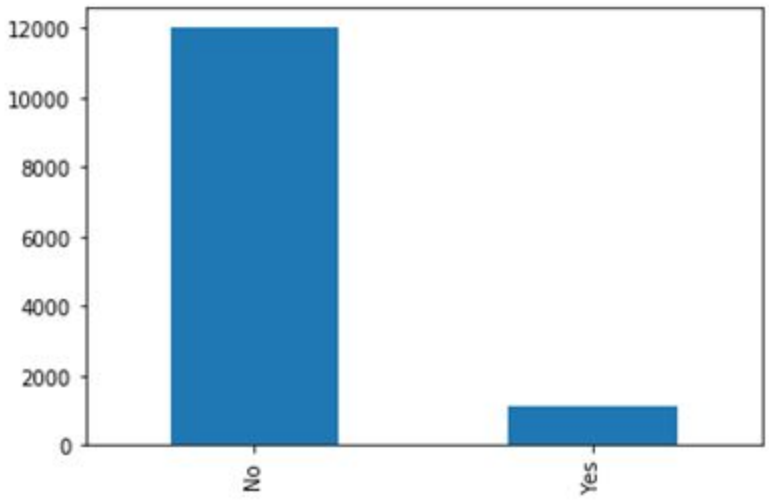
In [7]:
```
df
```

Out[7]:

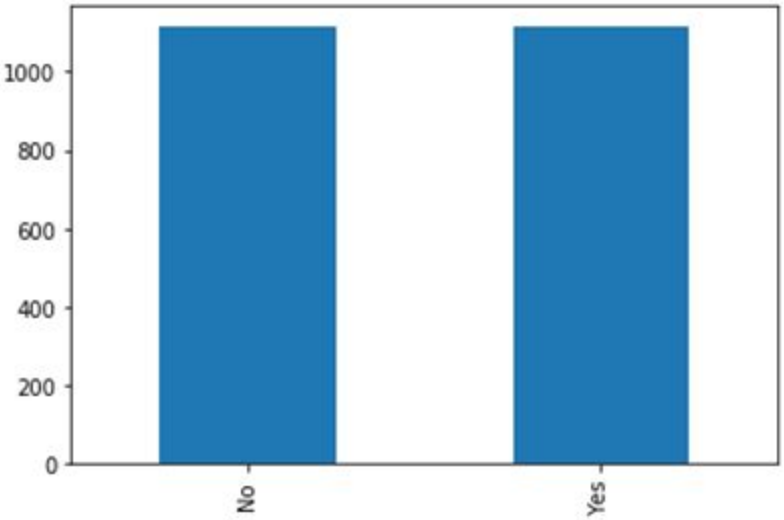
	Text	Bullying
0	what's your favorite song? :D A: I lik...	No
1	<3 A: </3 ? haha jk! <33	No
2	"hey angel you duh sexy" A: Real...	No
3	(: A: :(	No
4	**********MEOWWW**********...	No
5	any makeup tips? i suck at doing my makeup lol...	No

Target Variable Sampling:

Before we started our analysis, we faced an issue with our target variable. Majority of the entries in our dataset were appropriate comments which made our dataset unbalanced. With a potential low recall score, our results would be rather inaccurate.



Therefore, we subsetted our data frame by it's category and sampled our majority variable to have a 50-50 split.



Our final dataset consisted of 2226 rows and 2 columns with an equal 50-50 split of categories.

Sentence Cleaning:

Cleaning our sentences was a rather bigger challenge than expected. The reason being, unlike other social media data, our sentences were from records generated in 2010. Back then, some websites had multiple HTML tags - "
" which are used to generate a new line on the page. Therefore, we had to edit our defined function to remove these words for our analysis.

Furthermore, we also experienced a lot of words that were not included in the NLTK stopwords list due to spelling mistakes and short forms (for example, only the letter “u” was used for “you”). We edited our function to remove these words as well.

We also removed general stop words, symbols and converted the sentences to lowercase.

Our edited defined function was as follows:

In [76]:
```
def sent_to_words(raw_sent):
    letters_only = re.sub("[^a-zA-Z]", " ",raw_sent)
    words = letters_only.lower().split()
    stops = set(stopwords.words("english"))
    meaningful_words = [w for w in words if not w in stops]
    meaningful_words = [item.replace("br", "") for item in meaningful_words]
    meaningful_words = [item.replace("lol", "") for item in meaningful_words]
    meaningful_words = [item.replace("u", "") for item in meaningful_words]
    meaningful_words = [item.replace("im", "") for item in meaningful_words]
    meaningful_words = [item.replace("r", "") for item in meaningful_words]
    meaningful_words = [item.replace("would", "") for item in meaningful_words]
    return( " ".join( meaningful_words ))
```

In [45]:
```
def clean_sent_length(raw_sent):
    letters_only = re.sub("[^a-zA-Z]", " ",raw_sent)
    words = letters_only.lower().split()
    stops = set(stopwords.words("english"))
    meaningful_words = [w for w in words if not w in stops]
    return(len(meaningful_words))
```

Sentence Vectorization:

As our problem was a classification problem, to use classification models we had to vectorize our sentences. Our vectorization method used was Countvectorizer as we proceeded to generate our models.

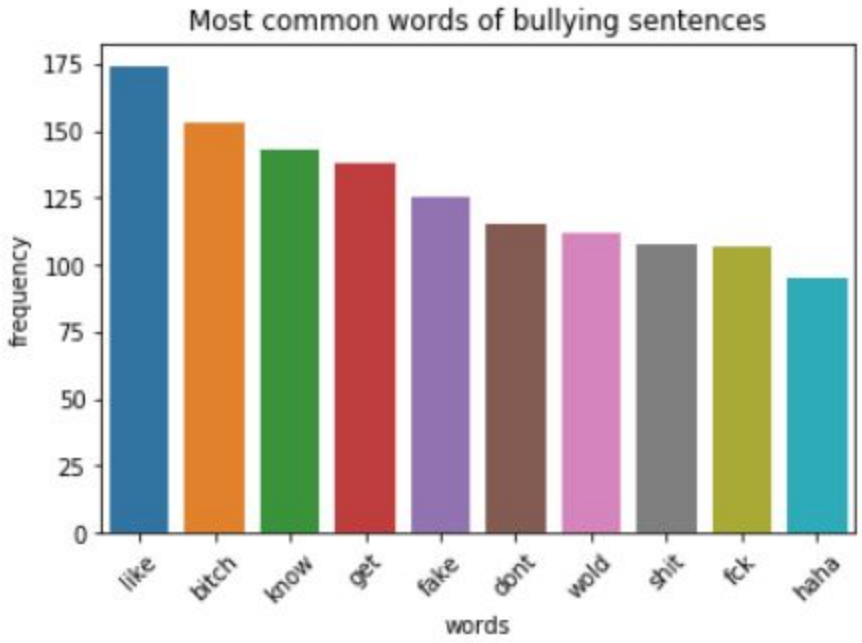
Exploratory Data Analysis:

Before we went on with model training and testing, we decided to do a visual analysis as we could determine some insights in our conclusions.

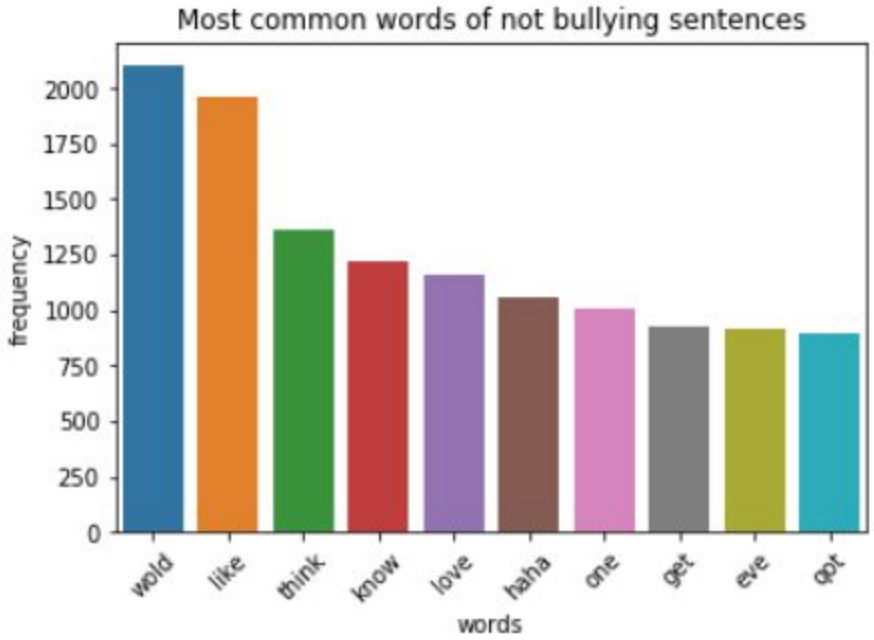
Word Frequencies of both Labels:

We conducted a word frequency analysis to determine if any topics could be generated which would lay a foundation to our classification rather than just training or testing our models.

The most common words for bullying sentences are as follows:



Similarly, we generated a frequency plot of the non-bullying sentences:



Word Cloud Analysis:

While generating most common words helps, we also went deeper into what these sentences actually meant as some of the words were very generic.

The word cloud for bullying sentences are as follows **(We scratched out the vulgar words as it would be inappropriate to show that in our submission)** :

Similarly, the word cloud for non-bullying sentences are as follows:

Model Training and Testing:

In our analysis, we built a total of **17 models** (7 with the balanced dataset, 7 with the unbalanced dataset and 3 with TF - IDF Vectorizer which will be discussed later).

We first went with **5 models** namely:

1. Gaussian Naive Bayes Classifier.
2. Decision Tree Classifier.
3. Logistic Regression.
4. K - Nearest Neighbours.
5. Random Forest Classifier.

We also later proceeded with **two more models** after our presentation:

1. Adaptive Boosting (AdaBoost) Classifier:
   This is another ensemble method which fitted a classifier on our dataset but also fitted additional weak classifiers to increase the overall model accuracy by combining them. Basically, it combines multiple classifiers to create on final strong classifier on our data. We decided to use this model as our ensemble models were the best performed ones.

2. Support Vector Machines:
   We decided to also try Support Vector Machines to see how to performs against the other classifier models.

Our parameters used in our classifiers are as follows:

```
In [99]: Classifiers = [
             LogisticRegression(C=10,solver='liblinear',max_iter=200),
             KNeighborsClassifier(3),
             SVC(kernel="rbf", C=0.025, probability=True),
             DecisionTreeClassifier(),
             RandomForestClassifier(n_estimators=200),
             AdaBoostClassifier(),
             GaussianNB()]
```

Model Evaluation:

Model Evaluation with Unbalanced Dataset:

With our unbalanced dataset, some of our models performed very well and majority of our accuracies were in the 90s range.

Models (Unbalanced Dataset)	Countvectorizer Accuracy
Naive Bayes	72.23%
Decision Tree Classifier	90.24%
Logistic Regression	91.30%
Support Vector Machines	**91.57%**
K - Nearest Neighbours	91.95%
Adaptive Boosting Classifier	**92.10%**
Random Forest Classifier	93.01%

We first started off with our Gaussian Naive Bayes classifier and received a 72% accuracy. While this is a decent accuracy, we knew our accuracy could increase. We then went in with Decision trees and our accuracy boosted to 90.24%. We also tried a simple Logistic Regression which brought our accuracy percentage even higher.

Our initial best models were K - Nearest Neighbours and our best performing model was the Random Forest Classifier.

Our new models ranked 2nd and 4th respectively which is a rather good performance as compared to the other models.

However, our Random Forest Model remained victorious when it came to our models performing against each other. The Adaptive Boosting Classifier performed good as well.

The high accuracy can be explained as since majority of our original dataset had non-bullying dataset, the prediction for that category was easier for the model. Therefore, these models cannot be exactly reliable for solving real-world problems.

Model Evaluation with Balanced Dataset

As compared to the results shown in our presentation, we used a different sampling technique as balancing and then concatenating the datasets were sorted by their category which is why our initial models were faulty. The accuracy received for our models with the balanced dataset were as follows:

Models (Balanced Dataset)	Countvectorizer Accuracy
Naive Bayes	63.23%
Decision Tree Classifier	70.63%
Logistic Regression	70.40%
Support Vector Machines	47.98%
K - Nearest Neighbours	61.66%
Adaptive Boosting Classifier	71.75%
Random Forest Classifier	74.67%

With the balanced dataset and with sampling done on the dataset as a whole, the overall accuracy of our models is rather lower. Although the accuracy of our models is lower, since our dataset is balanced, these models are more reliable. The accuracy trend of models is almost the same except the Supper Vector Machine model performed poorly with a 47.98% accuracy. Our Random Forest Model still performs the best and remains victorious with a 74.67% accuracy.

Conclusions and Insights:

Visual Analysis:

Although model performance is a great way to generate insights, we feel that our visual analysis equally helped as our data was social-media oriented.

Word Frequency Analysis:

From our most generated words, we can conclude that the sentences that were classified as bullying sentences had vulgar language and also had laughing expressions as well. This is a perfect example of internet trolls who are notorious for cyberbullying.

From our non-bullying sentences, we did have a few positive words but also a few common/unnecessary words which had spelling mistakes. We therefore went on with a deeper analysis with word clouds.

Word Cloud Analysis:

From our word cloud analysis, it can be indeed confirmed that sentences with vulgar language and laughing statements were the majority of bullying sentences. These words ranged from common vulgar language to homophobic and body shaming statements which we blurred out as we felt it was inappropriate to include those words in our report. We can also see a mention of Justin Bieber in the bullying sentences. This was not a surprise to us as back in 2010, Justin Bieber was widely disliked over the internet due to the backlash of one of his most popular songs and was used globally over the internet in derogatory statements.

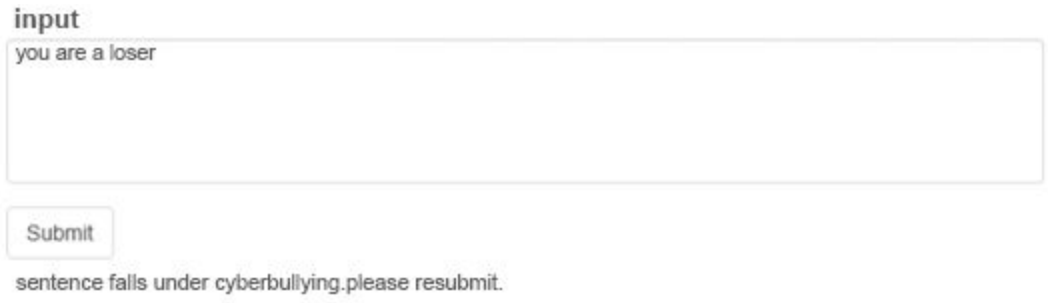
In our non-bullying sentences, we can see rather shallow topics (for example, fruits and driving) that were included.

Model Evaluation:

Our Random Forest Classifier was the best classification model that performed on our data. Although since our dataset just had a sentence and a label, there was no other feature importance that could be determined from it.

User Interface:

Given the time we had from the presentation to the report, we generated a simple User Interface that detects cyberbullying and requests the user to resubmit their entry. We used our Random Forest Model in our interface as that was our best performing model.



Other Findings:

Model Evaluation with TF-IDF Vectorizer:

We also decided to use a different vectorizer to see if our models perform better. We used the TF-IDF vectorizer and trained and tested 3 of our models again.

Models (Balanced Dataset)	TF-IDF Accuracy
Naive Bayes	61.61%
Logistic Regression	71.43%
Random Forest Classifier	65.18%

In this case, our best model is the Logistic Regression model and not the Random Forest Classifier. However, the accuracy of our best model with TF - IDF still does not match with the accuracy with our best model under Countvectorizer so we considered Countvectorizer in our main analysis.

However, which vectorization method is better and which model is the best is still subjective but from our insights, we have considered the Random Forest Classifier model with the Countvectorizer feature in our deployment.

Challenges Faced:

Our balanced dataset was rather small. Perhaps if the dataset was bigger, we could have extracted more features. The usage of Convolutional Neural Networks and Deep Neural Networks was not feasible as well due to the small size of our dataset.

The small dataset also resulted in most of our different models having almost similar accuracy whether it was balanced or not.

A potential solution in the future could be oversampling our dataset and using a model such as Isolation Forest.