**Biases in Toxic Comment Classification**

Rohan Alli, Kanak Sharma, Jerome, Aditya Sihag

***Abstract*— Importance of Toxic Comment classification has widely grown over the past decade due to the blast of information and the need to maintain civility in online expression. Comment classification has always been challenging with the continuous change in urban dictionary. Sentiment Analysis gives us better idea on the comments’ innate meaning. They have been tried and tested individually but not in combination. Firstly, we use a Machine Learning Model to classify the negative comments. Then, we use the combination of VADER (Valence Aware Dictionary and sEntiment Reasoner), a simple rule-based model for general sentiment analysis and Machine Learning Model. We find that there is an increase in accuracy and reduced bias towards negative words.**

**Introduction**

With much data being generated every day in Social Media, Toxic Comment Classification has grown complex. The complexity comes out due to growth of urban dictionary. The Urban dictionary uses cuss words both in a negative and positive fashion. Till now we have observed that many of the classifiers which currently exist are mostly biased towards Negative words as they have been trained on the words and used to identify negative/toxic words. These classifiers don’t consider the innate meaning of the sentence. Comments containing negative words may not turn out to be toxic comments after all. We rectify this bias by including deep sentiment analysis of the comments along with a normal classifier and make a combined model with respective weights. Then we quantify the bias by finding the difference between accuracies of our model which is the combination of regular classifier with sentiment analysis and the normal classifier.

Toxic Comment Classification builds a multi-headed model that is capable of detecting different types of toxicity like threats, obscenity, insult, etc. We use Binary relevance which is arguably the most intuitive solution for learning from multi-label examples. It works by decomposing the multi-label learning task into a number of independent binary learning tasks (one per class label). It has its potential weakness in ignoring correlations between labels, but we are not concerned with any correlation between labels. We choose SKLearn’s simple Logistic regression as a classifier and Stochastic Average Gradient as solver which is used over large datasets to achieve convergence faster by incorporating a memory of previous gradient values.

VADER sentiment analysis depends on a dictionary which maps lexical features to emotion intensities called sentiment scores. It returns a sentiment score in the range -1 to 1, from most negative to most positive. The sentiment score of the text is obtained by summing up the intensity of each word in the text.

After combination, we find that the average of outputs from the original classifier and the Vader are significantly more accurate than the original classifier itself. This Shows that the sentiment analysis plays an major role in the comment classification.

**Problem Statement**

Our problem statement is to identify the bias in the regular Machine Learning classifier for the toxic comment classification problem against negative words and mitigate them by using model which is a combination of factors from in-depth Sentiment analysis and regular Classifier.

**Related Work**

*1. Measuring and Mitigating Unintended Bias in Text Classification*: This paper introduces the mitigation, an unsupervised approach based on balancing the training dataset. They demonstrate that this approach reduces the unintended bias without compromising overall model quality.

*2. Challenges for Toxic Comment Classification:*

*An In-Depth Error Analysis by Betty van Aken et al* deals with multiple approaches for toxic comment classification and shows that the approaches make different errors and can be combined into an ensemble with improved F1- measure.

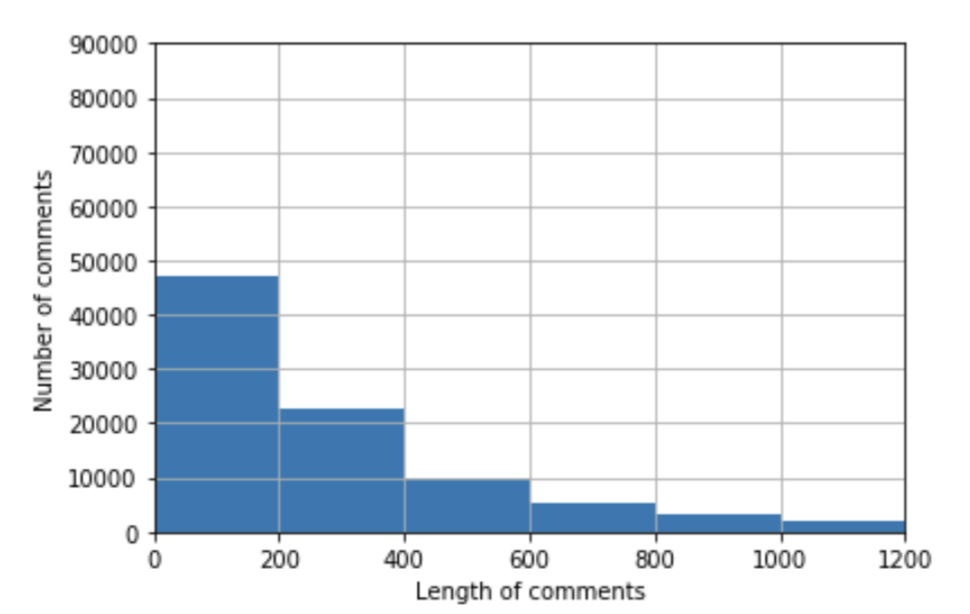
*3. VADER: A Parsimonious Rule-based Model for Sentiment Analysis of Social Media Text by C.J. Hutto and Eric Gilbert:* Introduced the VADER sentiment analysis method, and compare the accuracy with other present sentiment analysis methods.

**Dataset**

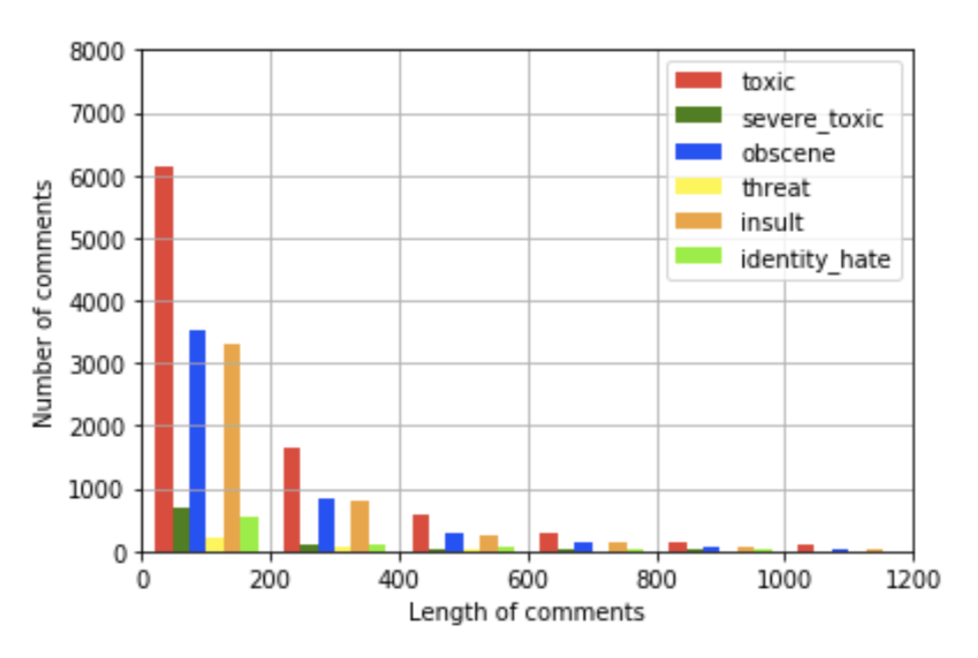
We have a dataset of a large number of Wikipedia comments which have been labeled by human raters (Turkers) for toxic behavior. This is available in the Kaggle website as part of Kaggle contest. The train.csv has 6 types of toxicity are: toxic, severe\_toxic , obscene, threat, insult, identity\_hate and an id : An 8-digit integer value, to identify the person who had written this comment. We have more than 160,000 labeled comments to train our model. Each column of type of toxicity contains the binary values of 0/1. We also have the test.csv which has around more than 60,000 comments which also have the correct binary values.

**Data Visualisation**

From the visualisation below, we can observe that comments have varying lengths from within 200 upto 1200. The majority of comments have length upto 200, and as we move towards greater lengths, the number of comments keeps on falling.

****

From the second visualisation below, we can observe the number of words falling under the six different outcome labels toxic, severe\_toxic, obscene, etc along with their lengths. Here also similar to the first plot, we observe that most of the abusive comments have lengths under 200, and this number falls with the length of comments.

****

**Data Preprocessing**

**1. Preparing a string containing all punctuations to be removed:**

The string library contains punctuation characters. This is imported and all numbers are appended to this string. We represent the punctuation by \’ instead of ‘. We also remove unwanted spaces and new line.

**2. Updating the list of stop words:**

Stop words are those words that are frequently used in both written and verbal

communication and thereby do not have either a positive/negative impact on our statement. E.g. ‘is’, ‘this’, ‘us’, etc. Even we remove single letter words which has no meaning.

**3. Stemming and Lemmatization** :

Stemming is the process of converting inflected/derived words to their word stem or the root form. Lemmatizing is the process of grouping together the inflected forms of a word so they can be analysed as a single item. This is quite similar to stemming in its working but differs. The wordnet library in nltk will be used for this purpose. Stemmer and Lemmatizer are also imported from nltk.

**4. Applying Count Vectoriser :**

Count Vectoriser is used for converting a string of words into a matrix of words with column headers represented by words and their

values signifying the frequency of occurrence of the word. It accepts stop words, convert to lowercase( set as true), and regular expression as its parameters. Here, we will be supplying our custom list of stop words created earlier and using lowercase option. Regular expression will have its default value.

**5. Splitting dataset into Training and Testing:** The healthy split would be ⅔ of dataset for training and remaining ⅓ for testing. We also test on the test.csv file.

**Approach**

For the Toxic Comment Classification, we use Binary Relevance with Logistic Regression as a classifier. Our approach is to consider the 6 labels are independent of each other and it is not our main motive to find the correlation between the labels. Therefore, we chose Binary Relevance. We choose simple Logistic regression as a classifier to get less accuracy so that we can mitigate the bias by showing the growth in accuracy rather than choosing a model with high accuracy which could overfit the data if we try to increase the accuracy more. We use Stochastic Average Gradient as our solver for the classifier. SAG method optimizes the sum of a finite number of smooth convex functions. Like stochastic gradient (SG) methods, the SAG method's iteration cost is independent of the number of terms in the sum. However, by incorporating a memory of previous gradient values the SAG method achieves a faster convergence rate than black-box SG methods. We predict the probabilities from the classifier for all the 6 columns, and to combine them into single value we take the average of these values.

For the Sentiment Analysis section in our proposed model, we use the VADER to address the sentiment of sentences. VADER is a lexicon and rule-based sentiment analysis tool that is specifically attuned to sentiments expressed in social media. It not only tells about the Positivity and Negativity score, but also tells us about how positive or negative a sentiment is. VADER combines the lexical features with consideration for five general rules that embody grammatical and syntactical conventions for expressing and emphasizing sentiment intensity. The result of VADER is effective and outperforms many of the present sentiment analysis methods available.

The basic operating principle of VADER is as follows: It first constructs and empirically validate a gold-standard list of lexical features (along with their associated sentiment intensity measures) which are specifically attuned to sentiment in microblog-like context; then VADER combines these lexical features with consideration for five general rules that embody grammatical and syntactical conventions for expressing and emphasizing sentiment intensity [5]. VADER keeps a total of 7,500+ lexical features as a human-validated gold-standard sentiment lexicon [6]. It then use data-driven iterative inductive coding analysis to identify generalizable heuristics for assessing sentiment in text. Lastly, VADER includes the impact of grammatical and syntactical rules into analysis process. Although it is not a typical machine learning approach, it still achieves higher accuracy than many well-established ML methods. Moreover, VADER does not severely suffer from a speed-performance tradeoff.

To use the VADER Sentiment Analysis, we installed the toolkit in python platform. VADER is used to test every comment sentence in the dataset and output the result of sentiment analysis. The result of the VADER method include four values: positive score (pos), negative score (neg), neutral score (neu), and compound score (compound). The positive, negative, and neutral scores are ratios for proportions of text that fall in each category (so these should all add up to be 1... or close to it with float operation). These are the most useful metrics if you want multidimensional measures of sentiment for a given sentence. The compound score is computed by summing the valence scores of each word in the lexicon, adjusted according to the rules, and then normalized to be between -1 (most extreme negative) and +1 (most extreme positive).

As we have the score of VADER and score retrieved from dataset label, we computed the accuracy of VADER by evaluating if a sentence has similar VADER score and dataset score. The evaluation procedure is further introduced in the next section.

At the end we incorporate the prediction from the sentiment analysis and the classifier above into one single prediction value using some means of weighted average.

**Evaluation**

The evaluation process of VADER section is straightforward. We count the number of sentences which have same results in VADER and sample dataset. The key point is the method to determine whether the VADER result is same as the sample dataset result. For a sample sentence, we define it as toxic comment if it has number of toxicity greater-than-or-equal-to half of the total number of toxicity, which is [3:6]. For the VADER part, we try different measurement standard, including: negative score, compound score, and (negative + neutral/2) score. If the negative score of a comment is greater-than-or-equal-to 0.5, we define it as toxic comment. If the compound score of a comment is less-than-or-equal-to -0.05, we define it as toxic comment. Lastly, the (negative + neutral/2) is greater-than-or-equal-to 0.65, we define it as toxic comment. The results of VADER accuracy on train dataset and test dataset is shown in the Table1 and Table2.

|  |  |  |
| --- | --- | --- |
| **Type** | **Num of Correct Classification** | **Total Accuracy** |
| Compound | 108,545 | 68.023% |
| Negative | 153,489 | 96.189% |
| Negative + (Neutral/2) | 152,805 | 95.759% |

**Table1: VADER result of Train.csv dataset (Total of 159,571 samples)**

|  |  |  |
| --- | --- | --- |
| **Type** | **Num of Correct Classification** | **Total Accuracy** |
| Compound | 43,662 | 68.245% |
| Negative | 60,973 | 95.303% |
| Negative + (Neutral/2) | 60,337 | 94.309% |

**Table2: VADER result of Test.csv dataset (Total of 63,978 samples)**

We initially run the basic Classifier of Logistic regression with Binary relevance on the train and the test datasets. The accuracies are given in the Table3.

|  |  |
| --- | --- |
| **Test Dataset** | **Accuracy** |
| Train.csv | 92.77% |
| Test.csv | 94.54% |

**Table3 : ML model accuracies.**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **DataSet** | **LR Value** | **Vader Value** | **Ratio** | **Accuracy** |
| Train.csv | Average | Negative | 1:1 | 97.09% |
| 2:3 | 97.27% |
| Average | Negative + (Neutral/2) | 1:1 | 97.05% |
| 2:3 | 97.12% |
| Test.csv | Average | Negative | 1:1 | 96.57% |
| 2:3 | 96.41% |
| Average | Negative + (Neutral/2) | 1:1 | 96.70% |
| 2:3 | 96.61% |

**Table4 : Final Model Accuracies**

Now we combine the values from both Vader and the ML model. We do not use compound value from the Vader as it give less accuracy, we only use the other two combinations with different ratios. The results are in Table4.

**Results**

We can see that there is an overall increase of 5% on an average in the Train.csv file and 2% in Test.csv. The increase is mostly due to the cases where the use of the Negative words from urban dictionary but in a positive meaning. In these cases our sentiment analysis kicks in and slightly reduces the overall negative coefficient of the comment sentence making it less toxic. There is no major change when we consider different ratios unless we go for larger ratio on vader values, which again will be a not appropriate method to do as we lose the significance of the classifier predictions

**Conclusion**

We have discussed the possible biases exist in the researches of toxic comment classification. Furthermore, we have

proposed a method to help mediate the biases. The results showed that the combination of Logistic Regression Toxic Comment Classifier and VADER sentiment analysis toolkit help improve the total accuracy of the baseline classifier. There are still different approaches we could try to further improve the result of the model. Moreover, the existence of biases in text classifiers is a phenomenon that cannot be neglected. We believe that the mediation of the biases in machine learning is going to be a popular topic in the future.

**References**

[1]https:/[/www.kaggle.com/c/jigsaw-toxic-](http://www.kaggle.com/c/jigsaw-toxic-) comment-classification-challenge/data [2]https://medium.com[/@nupurbaghel](about:blank)/toxic- comment-classification-f6e075c348

[[3] [5] [6] http://comp.social.gatech.edu/papers/icwsm1](http://comp.social.gatech.edu/papers/icwsm1)

4.vader.hutto.pd

[4]https://nlp.stanford.edu/pubs/sidaw12\_simplesentiment.pdf