UNIVERSITÀ
DEGLI STUDI
DI MILANO

Department of Economics, Management and Quantitative Methods

Hate Speech Detection on White Supremacist Forum Content

RIJIN BABY

954007

Data Science and Economics

Text Mining and Sentiment Analysis

Contents

List of Figures

Abstract

Hate speech detection on is critical for applications like controversial event extraction, building AI chatter-bots, content recommendation, and sentiment analysis. The complexity of the natural language constructs makes this task very challenging. The goal of the project is to create models for classifying text contents as offensive. The data is collected from a white supremacist forum and the overall analysis focus on the steps to deal with hate speech detection framework and the best machine learning models that deliver good results.

1. Introduction

Forum are medium that expresses increasing aspects of modern life, mediating large portions of it. While it equally empowers users to freely share opinions, it equally empowers them to restrict that very freedom in other users by facilitating anti-social behavior, trolling, bullying, harassment and hate speech. This misuse of freedom of speech has become a major issue not only on particular micro blogging platforms but the world at large: an important issue to tackle at organizational and national levels. Hate speech for example, can have consequences that go well beyond that individual grief due to online interaction, it can fuel real life violence In contrast to enabling free speech, the online forums presents a safe space for individuals to spread hate, whether towards individuals or groups [1, 2]. While platforms such as Facebook, Twitter and YouTube do provide mechanisms to report problematic content such as hate speech. These are barely adequate, because these methods of reporting are manual. As the users from diverse backgrounds use different words related to hate speech, it is difficult to detect this type hate speech content under the manual methods. Therefore,the task of hate speech detection remains challenging due to plenty of hateful content, the unavailability of benchmarks and lack of efficient approaches.

Automatic detection of humiliation, hate speech, abusive language and threats present important approaches for realizing this responsibility and protecting users, and have therefore come to dominate research areas. These areas are constituted by different sub areas. Sarcasm detection [3], sentiment analysis and hate speech [4] all focus on in-depth analysis of subjective language in online social networks (OSN). For our area of concern, hate speech, it is important to recognize that finding a complete and comprehensive definition or explanation of hate speech is a difficult task.

2. Data-set

Data files contain text extracted from Stormfront, a white supremacist forum. A random set of forums posts have been sampled from several subforums and split into sentences. Those sentences

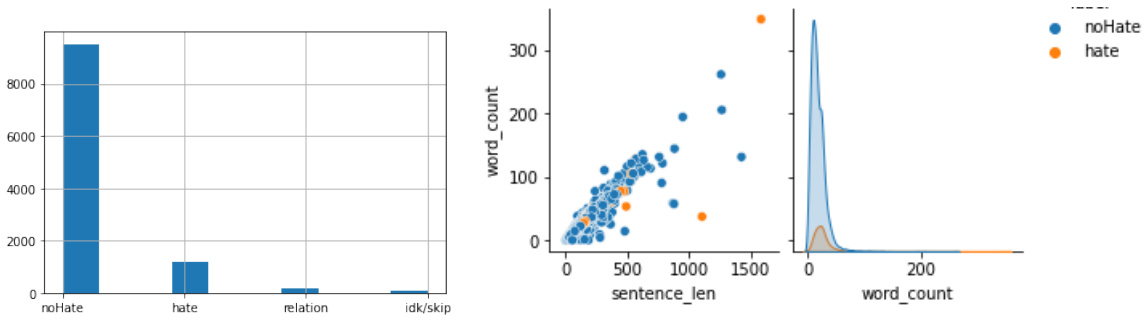

Figure 1: (a) Labels in the data-set (b) Word count and sentence length

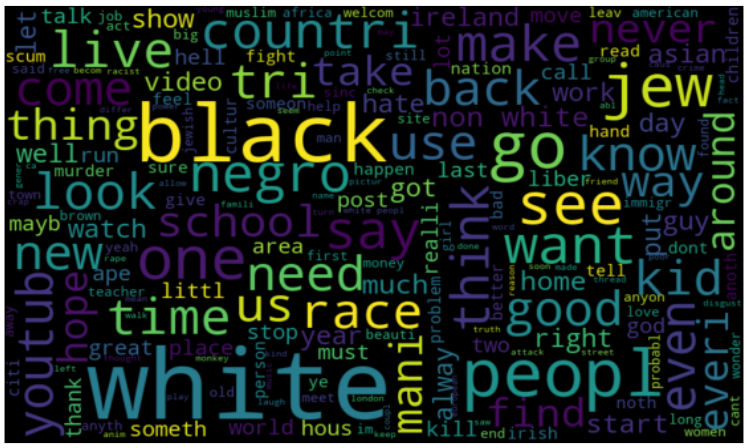

Figure 2: Most Frequent Words in Hate Speech Dataset

have been manually labelled as containing hate speech or not, according to certain annotation guidelines.Refer Github for more information regarding the data. On manual inspection on the data-set I found a few random texts that are classified as hate but they are actually not.

For this project we are ignoring the labels relation and idk/skip and consider only hate and nohate. Also, as shown in figure 4 the data-set is highly unbalanced. In the data-set repository there is also a sample train and test data-set which has a balanced distribution of hate and noHate classes.

3. Research Questions and Methodology

3.1. Research Questions

1. Determine the most relevant terminology for each category (hate and nohate)
2. Build models for classifying text contents as offensive or note

3.2. Methodology

Detection of hate speech is a challenging task in general hence, it is difficult to write the rules of hate speech hate speech detection by hand. Thus, we proposed a Hate–speech detection framework to

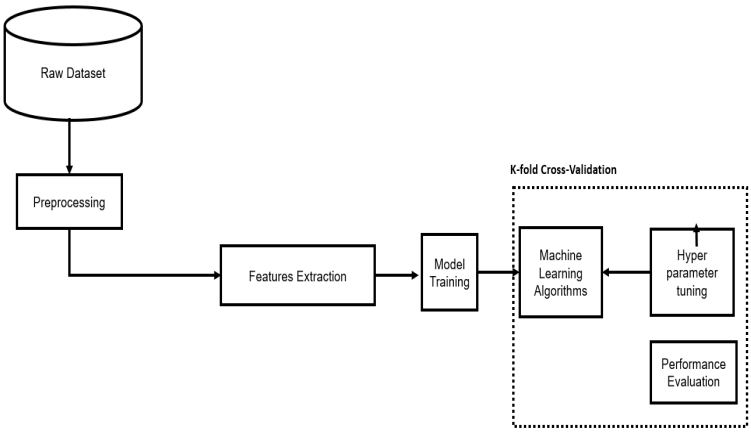

Figure 3: Hate Speech Detection Framework

detect hate speech on social media. Essentially it involves four phases which are: data prepossessing, features extraction, parameter tuning of machine learning model and performance evaluation of framework as shown in figure 3.

The data pre-process steps are pretty intense that include a variety of steps using NLTK library like

- Split into Words
- Normalize text
- Remove punctuation
- Filter out stop words
- Remove url, user tags
- Lemmatization and stemming

Since the data-set is highly unbalanced I tested multiple strategies like SMOTE oversampling, hybrid under-sampling [5] and random under-sampling As a starting in order to get some data-set insight (hate vs noHate) I will do the Log-likelihood test to extract the top words by category.

4. Top words using Log Likelihood

[6] It is always possible that the differences you have found between fiction and non-fiction are just a random, chance happening. The statistical term for this is "significance". If results are significant, we are reasonably certain (usually 95% certain, sometimes 99% certain) that these results are not due to chance. Very often, you will need to test whether your results are significant. Two common tests of significance are chi-square and log likelihood. We will use log likelihood in this session. The only information needed to do the log likelihood test is: - frequency in corpus 1 frequency in corpus

```
Top HATE tokens by likelihood:

black      56.427571
jews       54.884813
negro      44.174380
ape        33.346322
race       24.830260
scum       24.325840
mud        20.384637
non        20.364540
liberal    20.259751
white      19.448300
Name: llikelihood, dtype: float64
```

```
Top NOHATE tokens by likelihood:

youtube    34.542251
post       19.996576
meet       19.023056
pm         17.552700
sf         15.575919
music      13.569030
year       12.392726
march      12.087349
link       11.640213
video      10.407415
Name: llikelihood, dtype: float64
```

Figure 4: Likelihood Results

2 total number of words in corpus 1 total number of words in corpus 2. If the log likelihood for your result is greater than 6.63, the probability of the result - i.e. the difference between the two corpora - happening by chance is less than 1%. So we can be 99% certain that the result actually means something. This is usually expressed as $p < 0.01$. From our result shown in figure 4, we can see a qualitative distinction between the 2 groups. The 'hate' words align to what is expected

5. Model Results and Discussion

After performing pre-processing and features extraction, we move to our final experiments. The experiments are performed using the Scikit-learn. Scikit-learn presents variation of classifiers according to the group of the algorithm (e.g., rule-based, decision tree-based, etc.) We consider four well known machine learning algorithms used for hate speech detection: Logistic regression (LR), Linear Support Vector Machines (SVM), Random Forest (RF) and XGboost (XGB) Classifier. We performed 5 fold validation for the training of each model on Hate speech data-set. Measures that are considered are accuracy, recall, precision and F1 scores [7]

- Accuracy - Accuracy is the most intuitive performance measure and it is simply a ratio of correctly predicted observation to the total observations. One may think that, if we have high accuracy then our model is best. Yes, accuracy is a great measure but only when you have symmetric data-sets where values of false positive and false negatives are almost same.

- Precision - Precision is the ratio of correctly predicted positive observations to the total predicted positive observations. The question that this metric answer is of all text that got labeled as hate, how many were actually hate? High precision relates to the low false positive rate

- Recall (Sensitivity) - Recall is the ratio of correctly predicted positive observations to the all observations in actual class - yes. The question recall answers is: Of all the text that truly are hate, how many did we label correctly?

- F1 score - F1 Score is the weighted average of Precision and Recall. Therefore, this score takes both false positives and false negatives into account. Intuitively it is not as easy to understand as accuracy, but F1 is usually more useful than accuracy, especially if you have an uneven class distribution

Comparatively the classifiers showed improved results on the combination TF-IDF and uni-gram feature so using this feature we have calculated recall, precision and F-measure of all classifiers. This can be because of the short comment length, lack of structure and informality, together with the existence of diminutives and typos. Therefore, it is difficult to find a set of tokens that occur together.

Since the data-set is highly unbalanced I have tried 4 different approaches in creating the training data-set like (a)original data (b)over-sampled data (c)under-sampled data (d)hybrid-sampled data [5]

	Precision	Recall	F1 score	Accuracy
Logistic Regression	0.426	0.485	0.454	0.870
Random Forest	0.395	0.339	0.365	0.868
XGBoost	0.447	0.444	0.445	0.877
SVM	0.397	0.469	0.430	0.861

	Precision	Recall	F1 score	Accuracy
Logistic Regression	0.417	0.515	0.461	0.865
Random Forest	0.423	0.356	0.386	0.874
XGBoost	0.451	0.444	0.447	0.878
SVM	0.407	0.515	0.455	0.862

Figure 5: (a) Model performance using oversampling (b) Model performance using hybrid-sampling

As we can see in Figure 5 all the models gave a better accuracy but when it comes to other metrics like precision, recall and f1 score the performance is not up to the mark. Assuming the issue is caused by the unbalanced data-set that we have. And clearly all the sampling techniques applied didn't gave any boost to the model performance. Out of all modes only Logistic regression gave

good enough results compared to other models. This is shocking as there are couple of analysis available in the web using same data-set but they are actually performed on the sample train and test data that is available in the data-set repository.

I was curious and used the sample train and test data-set which is balanced and has lesser observations. I followed the same pre-processing and modeling steps and now there is a satisfying performance results which are similar to the ones available. The performance shown in the table 1 can be considered as a proof that this framework can deliver better results if trained by an excellent data-set.

Table 1: Precision, Recall, F1-Measure and Accuracy of Classification using different classifiers

Classifiers	**Hate Class**			**Non-Hate Class**			**Accuracy**
	Precision	**Recall**	**F-Measure**	**Precision**	**Recall**	**F-Measure**	
LR	0.67	0.76	0.71	0.72	0.64	0.68	70.08
RF	0.75	0.64	0.69	0.69	0.79	0.74	**72.17**
XGB	0.78	0.52	0.63	0.64	0.86	0.74	69.24
SVM	0.68	0.77	0.72	0.74	0.64	0.68	70.71

6. Conclusion

This study aim is to detection of hate speech using machine learning. To achieve this, we proposed a framework. A set of experiments were performed to measure the effectiveness of framework. We have compared, Random Forest (RF), Linear Support Vector Machine (SVM), XGBoost and Logistic Regression (LR) on sets of model hyper-parameters and features values. Among different machine learning algorithms, RF produced the best results when using the combination of the features: TF-IDF and unigram. Furthermore, We have learned from this study that most of the text which are subjected to hate speech are based on the race, religious, racism, nationality and color.

Another main concern is with the Stormfront dataset which is unbalanced, large majority text with few words and short sentence. Hence the model will not perform with additional data coming from external sources. I can't claim that my framework would be the best based on the model results, but I am confident and will test if the framework can deliver better results using a unbiased data-set.

As a future score we need to train the framework with an improved data-set from multiple sources and has variety of contents and most importantly with correct labels. The python notebook to reproduce all of my work can be found in my personal GitHub repository.

References

[1] ADL, Adl report.
URL https://www.adl.org/resources/reports/murder-and-extremism-in-the-united-states-in-2017

[2] S. Naganna Chetty, Hate speech review in the context of online social networks.
URL https://www.sciencedirect.com/science/article/abs/pii/S1359178917301064

[3] A. A. A. N. LE HOANG SON, AKSHI KUMAR, M. ABDEL-BASSET, Sarcasm detection using soft attention-based bidirectional long short-term memory model with convolution network.
URL https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=8641269

[4] M. B. Hajime Watanabe, T. Ohtsuki, Hate speech on twitter a pragmatic approach to collect hateful and offensive expressions and perform hate speech detection.
URL https://www.researchgate.net/publication/323205135_Hate_Speech_on_Twitter_A_Pragmatic_Approach_to_Collect_Hateful_and_Offensive_Expressions_and_Perform_Hate_Speech_Detection

[5] Overcoming class imbalance using smote techniques.
URL https://www.analyticsvidhya.com/blog/2020/10/overcoming-class-imbalance-using-smote-techni

[6] Testing for significance: Log likelihood.
URL https://www.lancaster.ac.uk/fss/courses/ling/corpus/blue/l08_4.htm

[7] Interpretation of performance measures.
URL https://blog.exsilio.com/all/accuracy-precision-recall-f1-score-interpretation-of-performa