Twitter Hate Speech-A pragmatic approach to detect hate speech using binary text classification

Project Final Report -2020

mumar-mumar@andrew.cmu.edu

msakir-msakir@andrew.cmu.edu

Abstract

Online Hate speech is growing rapidly with the growth and popularity of social networks and microblogging websites. Hate speech detection is critical for applications like controversial event extraction, building AI chatterbots, content recommendation, and sentiment analysis. Therefore, arises the necessity to detect such speech automatically content that presents hateful language or language inciting to hatred. In this paper, we discuss evaluation of several machine learning models to detect hate expressions on Twitter. Our experiments are based on a dataset containing 31963 tweets. Our approach is based on collecting features from the training set. These features are later used (in their vectorized form) to train machine learning algorithms. Our approach also uses a bidirectional Recurrent Neural Network (RNN) model. Our experiments show that our approach reaches an accuracy of 95% for Naïve bayes on detecting whether tweet is 'Hate' or 'non-Hate' (binary classification), an accuracy of 96% for linearSVC, and an accuracy of 94.5% for bidirectional RNN.

1 Introduction

With the rapid growth of social networks and microblogging websites, communication between people from different cultural and psychological backgrounds became more direct, resulting in more and more "cyber" conflicts between these people. Consequently, hate speech is used more and more, to the point where it became a serious problem invading these open spaces. Hate speech refers to the use of aggressive, abusive, violent, or offensive language, targeting a specific group of people sharing a common property, whether this property is their gender (i.e., sexism, homophobia), their ethnic group or race (i.e., racism) or their believes and religion, etc. While most of the online social networks and microblogging websites forbid the use of hate speech, the size of these networks and websites makes it almost impossible to control all their content. Therefore, arises the necessity to detect such speech and filter content that presents hateful language or language inciting to hatred. The manual way of filtering out hateful tweets is not scalable, motivating researchers to identify automated ways.

There are several techniques used for automated detection of hate speech. Even though most of the solutions for automated detection of offensive text rely on Natural Language Processing (NLP) approaches, there have lately been a tendency towards employing pure machine learning techniques like neural networks for that task (Pitsilis, 2018). Our experiments combine several NLP techniques (sentiment analysis, binary text classification) to train several machine learning and a deep learning model. Our approach lies on extracting textual features from training data and using Count Vectorizer and TFIDF transformer to use vectorized form of the features. These features are used to train a baseline model of Naïve Bayes algorithm. Baseline model is evaluated against our 1st approach of linearSVC machine learning algorithm, along with our 2nd approach of bi-directional Recurrent Neural Network Deep learning model.

The remainder of the paper is structed as follows: in Section 2 we discuss some of the related work, in Section 3 we discuss what and how data was used in this project, in Section 4 we discuss the complete methodology that was adopted to solve the task along with the description of baseline and actual models used, in Section 5 we discuss evaluation and results of our experiments and finally in Section 6 we conclude our project.

2 Related Work

The analysis of subjective language on social networks has been deeply studied and applied on different fields varying from sentiment analysis to sarcasm detection or detection of rumors etc (Watanabe and Ohtsuki ,2018). There is also some work done that addresses detection of Twitter hate speech using automated methods. Pitsilis and Langseth (2018) did multiclass classification (Neutral, Racism, Sexism) using different types of classifiers. They reached a score of 92% for Accuracy, Recall and F1 using single class classifier. Using ensemble classifier, they reached a score of 92% accuracy and 93% for recall and F1. They also implemented an LSTM RNN which gave scores of 93% for accuracy, recall and F1. Watanable and Othsuki (2018) also did multiclass classification (offensive, clean, hateful) and worked on extracting different set of features from their dataset. Using Random forest classifier, they got scores of 60%, 59% and 60% for accuracy, recall and F1. Using SVM, they got scores of 64%, 57% and 60% for accuracy, recall and F1. Using J48graft, they got scores of 79%, 78% and 78% for accuracy, recall and F1. They also did binary classification (clean and offensive) using all their features (sentiment based, semantic, unigram, pattern) combined. They got scores of 88% for precision, recall and F1 while testing on combined features. Badjatiya and Gupta (2017) also studied Twitter hate speech detection using different models. Their baseline method consisted of char-n grams, TFIDF and BOW (Bag of Words) approach. Using these different methods, they got scores of approximately 80% for Precision, Recall and F1. Their deep learning models consisted of CNN, and LSTM methods for which they got scores of 86% for precision, recall and F1 (CNN) and 85% for precision, recall and F1 (LSTM). Waseem and Hovy (2016) also implemented a logistic regression model using different set of features. Their scores were approximately 73% for F1 and precision and 78% for Recall.

3 Data

Our dataset consisted of 31963 annotated tweets collected from Kaggle **.** The dataset consists of 3 columns: tweets, tweet label (0 as non-hate and 1 as hate speech), and tweet id. The dataset was divided into 80% (25569) tweets for train data and 20% (6394) tweets for Test data using sklearn train-test split method. For bidirectional RNN model, dataset was divided into 85% for train data and 15% for test data. 10% of this train data was reserved for validation data. Our dataset contained approximately 90% tweets labelled as 0 (non-hate) which indicates that our dataset was highly biased towards the non-hate tweets.

For pre-processing, **Tweet-preprocessor** library was used. This library was used to remove URLs, user mentions (@) and numbers from tweets. Our datasets contained emojis in the form of symbols and these symbols were also improved by the library. Stopwords were also removed using nltk library. Using our pre-defined function (selfPreprocess), we removed single letters, # symbols from hashtags and expanded words using contraction mapping (for instance expanding I’ve to I have).

4 Methodology and Model description

Our approach focused on extracting features from our train data and passing these features as input to each model. For feature engineering, we worked on a set of different features. Not all these features were passed to the training algorithm because some features were more relevant and improved the score. The first set of features we worked on are the meta features. Meta features are used to extract general properties of the dataset that will help characterize the dataset. Our data frame consisted of ‘punctuation count’, ‘word count’ and the ‘unique word count’ as meta features. These features were extracted using our defined functions. ‘Polarity’ of each tweet was also calculated using TextBlob library. POS tags for each word in the tweet were also calculated using ‘pos_tag_sents’ function form nltk library. These tags were stored in the form of sentence in our dataframe where each row of POS_tagged column corresponds to POS tags of a tweet. (For example, Thank You will be stored as NN NN). Cosine similarity was also calculated for each tweet. The similarity was calculated using a pre-trained word2Vec model ‘glove-wiki-gigaword-300’. The model does not contain all the words that were part of our corpus and hence we dropped those words that are not part of trained model vocab. This was done using our own defined function. We also defined a set of keywords that typically represent hate-speech according to our dataset. These keywords were stored in a sentence (this sentence was later splitted into list of words when calculating similarity). N_similarity function from word2vec was used to calculate similarities between each tweet and the list of hate speech keywords.

We passed cleaned tweets, POS_tags, polarity, and the set of meta features as input to our model. To do so, we gave features as input while doing the train_test split using sk learn train_test split methods. These features were converted to their vectorized form using CountVectorizer and TfidfVectorizer. These vectorized form of features were then passed to the training models.

To solve our task, we used Multinomial Naïve Bayes as a baseline method. For multinomial NB, the set of features were converted to vectorized form using CountVectorizer (we also tried Tfidf vectorized form for multinomial NB but it gave us low scores, so we went with CountVectorizer instead). For our actual models, we used 2 approaches. Our 1st approach used linearSVC machine learning classifier. Our 2nd approach used bi-directional RNN deep learning method. For linear SVC, the set of features were converted to vectorized form using TfidfVectorizer. For bidirectional RNN, we used ‘sigmoid’ as activation function, 2 dense layers were added and a Dropout of 0.2 was used. Padding was also done using max sequence length of 100. Model was trained on 50 epochs with Early Stopping added. Our actual models were compared against our baseline methods and their scores and evaluation are discussed in the next section.

5 Evaluation and Results

For the evaluation we used standard metrics for classification accuracy, suitable for studying problems such as hate speech detection. In particular, we used Accuracy which is correctly predicted data points out of all data points. We also used Precision and Recall, with the former calculated as the ratio of the number of tweets correctly classified to a given class over the total number of tweets classified to that class, while the latter measures the ratio of messages correctly classified to a given class over the number of messages from that class. Additionally, the F-score is the harmonic mean of precision and recall, expressed as $F = 2 * (P * R) / (P+R)$. Results for baseline method and our 1st approach are given in Table 1. Results for our 2nd approach are given in Table 2.

	Accuracy	Precision	Recall	F1
Multinomial NB(Baseline)	0.9506	0.9916	0.2725	0.4275
LinearSVC	0.9640	0.8328	0.5866	0.6883

Table 1: Comparison of Baseline and linearSVC (1st approach)

	Accuracy	Precision	Recall
Bi-directional RNN	0.9449	0.9449	0.9449

Table 2: Evaluation of bi-directional RNN (2nd approach)

Based on evaluations of our models, we can clearly see that overall, our main models performed better than our baseline method. Our deep learning model, as expected, had the most consistent and better scores.

In the Machine learning models, the precision scores were very high whereas the recall was low. This could indicate that whatever the models predicted as hate speech, most of them were actually hate-speech. On the other hand, the model was not able to detect a good number of actual hate-speeches. This occurred due to the bias in the dataset. The machine learning models were not able to learn much about the hate speeches and thus could not identify most of them.

6 Conclusion

In this paper, we investigated the application of using machine learning and deep learning models for the task of hate speech detection. We found the scores to be consistent for deep learning model. Scores for accuracy and precision for our machine learning models were probably high because of the biased nature of our dataset. In general, scores for our models were quite better and are comparable to existing methods and research work. In future, we can improve this experiment by adding more relevant features and evaluate more sophisticated machine and deep learning models.

References

[1] Watanabe, Hajime & Bouazizi, Mondher & Ohtsuki, Tomoaki. (2018). Hate Speech on Twitter A Pragmatic Approach to Collect Hateful and Offensive Expressions and Perform Hate Speech Detection. IEEE Access. PP. 1-1. 10.1109/ACCESS.2018.2806394.

[2] Badjatiya, Pinkesh & Gupta, Shashank & Gupta, Manish & Varma, Vasudeva. (2017). Deep Learning for Hate Speech Detection in Tweets. 10.1145/3041021.3054223.

[3] Pitsilis, G.K., Ramampiaro, H. & Langseth, H. Effective hate-speech detection in Twitter data using recurrent neural networks. *Appl Intell* **48,** 4730–4742 (2018). https://doi.org/10.1007/s10489-018-1242-y

[4] Waseem, Zeerak & Hovy, Dirk. (2016). Hateful Symbols or Hateful People? Predictive Features for Hate Speech Detection on Twitter. 88-93. 10.18653/v1/N16-2013.