Hate Speech Detection

Janam Vaidya(201701027) | Akshar Joshi(201701123) | Shobhit Gola(201701129)

Abstract

In recent years, an increasing amount of hateful information spreading on social media platforms encouraged ethical communities, governments, and digital media companies to take countermeasures. Much research is being done in the area of automated hate speech detection online. The main objective of this research is to classify content in a hateful and non-hate speech in the area of racism, religion. However, we notice a significant difference between the performance of the two (i.e., non-hate vs. hate). Here we are comparing various ML and deep learning-based models to evaluate performance on tweets classification on hate and non-hate full. Our methods are evaluated on the largest collection of hate speech datasets based on Twitter. We found that how the simplest model of logistic regression can outperform deep learning-based LSTM model for hate speech classification.

Introduction

Term 'hate speech' was formally defined as 'any communication that disparages a person or a group on the basis of some characteristics such as race, color, ethnicity, gender, sexual orientation, nationality, religion, or other characteristics'.
Hate speech identification process is labor intensive, and time consuming. There is huge demand of an algorithm which can automate process with higher accuracy for hate speech identification. Companies like Twitter, Facebook, and Google are investing lots of resources for this objective and still being criticized for not having enough control for hate speech detection. Although current methods have reported promising results, we notice that their evaluations are largely biased towards detecting content that is non hate, as opposed to detecting and classifying real hateful content.

Overview

Our end goal is to classify tweets into two categories, racist/sexist, and neither. This paper will be training datasets of tweets on some classic Machine Learning models and a Deep Learning model. We will observe these experiments and make a note of which model performs better in the specific lot. There will also be a pinch of NLP involved in the process since we will be dealing with raw textual data. Since there will be a lot of textual information, that is tweets; we will use text classifiers.

Dataset

All data used for training and testing has been retrieved from Kaggle. As social media platforms are considered the place filled with all the hate speech, yet most of them, such as Twitter, have a very restrictive data distribution policy. These tweets were available publicly elsewhere. The tweets in the training dataset have three columns (id, label, and tweets). For simplicity, we say a tweet contains hate speech if it has a racist or sexist sentiment associated with it. The label, in the training dataset, will be '1' if the particular tweet is racist or sexist, and '0' if the tweet is neither. Also, the user's username in the tweets will be replaced with '@user.' An example of the data will be shown in the table below, i.e., the Training dataset. The dataset is not a very complex one. The training dataset and the testing dataset consist of 31,962 tweets and 17,197 tweets, respectively. The test dataset consists of id, target label (i.e., predicted after training the dataset on various models), and tweets.

Training Dataset

id	label	tweet
1	0	@user when a father is dysfunctional.. .dysfunction.#run
2	0	@user @user thanks for #lyft credit...wheelchair vans in pdx. #disappointed
3	0	bihday your majesty
14	1	@user #cnn calls #michigan middle school 'build the wall' chant " #tcot

Methodology

We first performed Text Cleaning and Preprocessing to remove irrelevant data and then text classification to categorize text into organized groups. Following this, we trained the dataset on various models and we compared the result and concluded which dataset gives the better accuracy in detecting hate speech.

Text Cleaning and Preprocessing

Text data needs to be cleaned and encoded to numerical values before giving them to Machine Learning and Deep Learning models, this process of cleaning and encoding is called as text cleaning and preprocessing. Text cleaning is basically an amalgamation of removing punctuations, unwanted tags, i.e. noise removal, expanding contractions, tokenization, converting all text to lower case, removing stop words, performing stemming, and performing lemmatization.

Text Classifier: TF-IDF

The main purpose of text classification is to convert open-ended text into categories. By vectorizing texts, we can further perform multiple tasks such as finding relevant documents, ranking, clustering, and so on.
TF-IDF stands for Term Frequency - Inverse Document Frequency. TF-IDF as a text classifier is used to calculate weight or score of words in the document or in a set of texts.
Term-Frequency(TF) of a word is more if the word has been used more number of times. TF for a word can be calculated by the formula: **tf(w,d) =count of w in d / number of words in d**
Inverse Document Frequency(IDF) is the inverse of the document frequency which measures the informativeness of word w. IDF for a word can be calculated by the formula:
idf(w) =N / occurrence of w in documents.
When we multiply tf(w,d) and idf(w), we will get the importance of a word in a text, relative to the set of texts.

w=word | d=document | N=count of corpus(set of texts)

Baseline Models

Multinomial Naïve Bayes

This technique is a generative model. It is a probabilistic algorithm based on the assumption that features of a class are independent. Bayes theorem calculates probability P(c—x) where c is the class of the possible outcomes and x is the given instance which has to be classified, representing some certain features.

$$P(c|x) = P(x|c) * P(c) / P(x)$$

Tag of a text is predicted by Naive Bayes, and it calculates the probability of every tag in a given text and its output is the tag which has the highest probability.

Random Forest

Random forest is a supervised learning algorithm. It builds an ensemble of decision trees with the help of the bagging method. The general idea of the bagging method is that many relatively uncorrelated decision trees operating as a committee will outperform any individual constituent models. Random forest builds multiple decision trees and merges them to get a more accurate and stable result.

Long Short-Term Memory

It is special kind of recurrent neural network that is capable of learning long term dependencies in data. This is achieved because the recurring module of the model has a combination of four layers interacting with each other.

Proposed Model

Since the dataset can be classified as racist/sexist and neither using '1' and '0', it can be seen as a basic classification problem. Thus, **Logistic Regression** would be a better way to predict probabilities. This has been tested in this paper. The reason why Logistic Regression has outperformed some state-of-the-art techniques, is because sometimes other factors such as the complexity of the dataset plays a huge role in the training of models. Logistic Regression uses the log odds ratio rather than probabilities and an iterative maximum likelihood method. The way Logistic Regression works is that it fits a single line to divide the space into two and so performs better than a decision tree when the data is distributed in a fashion such that it can be linearly classified. Representation of Logistic:

$$\text{Logistic}(\eta) = 1/(1 + \exp(-\eta))$$

Results and Analysis

Results

Experiments	F1-Score	Accuracy
Multinomial NB	19.56	93.51
Random Forest	48.91	95.00
LSTM	58.66	94.99
Logistic Regression	64.92	**96.13**

Conclusions

With the help of this paper, we detect racist/sexist tweets on twitter. We have observed that it's not always that state-of-the-art deep learning models outperforms all other models. Sometimes classic Machine Learning models perform better than deep learning models. The less complex the dataset, using classic machine learning models give better results. Also, when the TF-IDF text classifier was used, it gave better performances combined with Logistic Regression. A logistic regression model for binary classification can be far better than a neural network model because neural networks are more challenging to train and are more prone to overfitting than logistic regression.

Acknowledgements

I am grateful to Professor Priyanka Singh for providing me this opportunity to work under her and for guiding me throughout the internship. I would also like to thank Mr. Prashant for helping me along the way and clearing my doubts.