Machine Learning Engineer Nanodegree

Capstone Proposal

Raghunadh Chilukamari
February 25th, 2019

Twitter Sentiment Analysis

Domain Background

Hate speech is an unfortunately common occurrence on the internet. Often social media sites like Facebook and Twitter face the problem of identifying and censoring problematic posts while weighing the right to freedom of speech. The importance of detecting and moderating hate speech is evident from the strong connection between hate speech and actual hate crimes. Early identification of users promoting hate speech could enable outreach programs that attempt to prevent an escalation from speech to action. In spite of these reasons, NLP research on hate speech has been very limited, primarily due to lack of general definition of hate speech, an analysis of its demographic influences and an investigation of the most effective resources.

Problem Statement

The objective here is to detect hate speech in tweets. we say a tweet contains hate speech if it has a racist or sexist sentiment associated with it. So, the task is to classify racist or sexist tweets from other tweets. Given a training sample of tweets and labels, where label '1' denotes the tweet is racist/sexist and label '0' denotes the tweet is not racist/sexist, the goal is to predict label on the test dataset.

Datasets and Inputs

We will use the dataset provided in this competition for this problem. Datasets are described below

1. **train.csv** - For training the models, we provide a labelled dataset of tweets. The dataset is in the form of a csv file with each line storing a tweet id, its label and the tweet.
2. **test.csv** - The test data file contains only tweet ids and the tweet text with each tweet in a new line.

Solution Statement

This is a classification problem under supervised learning as we have to predict whether a tweet contains hate speech or not. we will clean the data, create the features, train various classification algorithm using the training data, select the model with best accuracy and further fine tune it to improve the model performance.

Benchmark Model

We will use below paper to benchmark our model:

https://pdfs.semanticscholar.org/a050/90ea0393284e83e961f199ea6cd03d13354f.pdf

Classifier	Training Corpus	Test Corpus	$F1_{pos}$	$F1_{neg}$	$F1_{neutral}$	F1
SVM	SB10k	SB10k	66.16	47.80	81.32	56.98
CNN	SB10k	SB10k	71.46	58.72	81.19	**65.09**
SVM	SB10k	MGS	49.50	38.62	66.41	44.06
CNN	SB10k	MGS	50.41	44.19	71.81	**47.30**
SVM	SB10k	DAI (full)	62.30	61.40	81.22	**61.85**
CNN	SB10k	DAI (full)	62.79	58.43	79.92	60.61
SVM	MGS	SB10k	67.77	53.23	80.20	60.50
CNN	MGS	SB10k	63.94	58.21	70.66	**61.07**
SVM	MGS	MGS	60.34	56.48	69.31	58.41
CNN	MGS	MGS	61.49	58.12	68.62	**59.80**
SVM	MGS	DAI (full)	59.32	56.03	74.83	57.68
CNN	MGS	DAI (full)	61.01	55.74	76.88	**58.38**

We can treat this benchmark as hypothesis as the training data used is different. we will take minimum F1 score 44.06 as benchmark to our model.

Evaluation Metrics

The metric used for evaluating the performance of models would be F1-Score.

The metric can be understood as -

True Positives (TP) - These are the correctly predicted positive values which means that the value of actual class is yes and the value of predicted class is also yes.

True Negatives (TN) - These are the correctly predicted negative values which means that the value of actual class is no and value of predicted class is also no.

False Positives (FP) – When actual class is no and predicted class is yes.

False Negatives (FN) – When actual class is yes but predicted class in no.

Precision = TP/TP+FP

Recall = TP/TP+FN

F1 Score = 2*(Recall * Precision) / (Recall + Precision)

Project Design

Data inspection:

> we will inspect the data for missing information such as tweets, labels, in both training and test datasets and check the distribution of labels in training dataset

Data cleaning:

> we will clean the data by removing stop words, removing punctuations, and stemming the data.

Visualization:

> we will get general understanding of the data by visualizing common words, words related to hate speech using word cloud.

Feature extraction:

> we will implement bag of words to extract features from data.

Model selection and tuning:

> We will try various models like Naive Bayes, SVM, Random Forest etc., pick the best model and try to fine tune the model by selecting best hyperparameters leveraging techniques like Grid Search.

Test and report:

> We will test the selected model on test data set and report accuracy scores.