Identifying and Categorizing Offensive Language in Social Media

Definition

Project Overview

Offensive language is pervasive in social media. Individuals frequently take advantage of the perceived anonymity of computer-mediated communication, using this to engage in behavior that many of them would not consider in real life. Online communities, social media platforms, and technology companies have been investing heavily in ways to cope with offensive language to prevent abusive behavior in social media.
One of the most effective strategies for tackling this problem is to use computational methods to identify offense, aggression, and hate speech in user-generated content (e.g. posts, comments, microblogs, etc.). This topic has attracted significant attention in recent years as evidenced in recent publications (Waseem et al. 2017; Davidson et al., 2017, Malmasi and Zampieri, 2018, Kumar et al. 2018) and workshops such as ALW and TRAC.

Problem Statement

Judge on whether a tweet is offensive or not. Please note that the data contains offensive or sensitive content, including profanity and racial slurs.

I will solve the first task (sub task A) only. Which is caring about detecting whether a tweet is offensive or not. It is simply a classification problem, can be solved using (Logistic Regression, Random Forest, KNN and SVM) . I will solve it using them all and then compare between them using confusion matrix, f1 score, pression, recall and accuracy. Using combination of preprocessing methods first (tokenization, stopwords removal and lemmatization).

- Tokenization is the method of breaking the text into smaller components (words and sentences.
- *Stopwords* removes stopwords from text (e.g. removes ‘and’, ‘or’, ‘in’...).
- *Lemmatization* usually refers to doing things properly with the use of a vocabulary and morphological analysis of words, normally aiming to remove inflectional endings only and to return the base or dictionary form of a word, which is known as the *lemma* . If confronted with the token *saw*, stemming might return just *s*, whereas lemmatization

would attempt to return either *see* or *saw* depending on whether the use of the token was as a verb or a noun.

Then I use vectorization (count and word2vec).

- Count

  CountVectorizer to learn the vocabulary of a set of texts and then transform them into a dataframe that can be used for building models.

- word2vec

  Word2vec is not a single algorithm but a combination of two techniques – CBOW(Continuous bag of words) and Skip-gram model. Both of these are shallow neural networks which map word(s) to the target variable which is also a word(s). Both of these techniques learn weights which act as word vector representations.

Then classify the tweets.

Metrics

- Confusion matrix can be used to represent TP, FP, TN and FN.
- F1 score can be computed for all classifiers. Which is used to measure a test's accuracy. The greater the f1 score, the better is the performance of our model.

$$F1 = 2 * \frac{1}{\frac{1}{precision} + \frac{1}{recall}}$$

F1 Score

- Precision : It is the number of correct positive results divided by the number of positive results predicted by the classifier.

$$Precision = \frac{TruePositives}{TruePositives + FalsePositives}$$

- Recall **:** It is the number of correct positive results divided by the number of ***all*** relevant samples (all samples that should have been identified as positive).

Analysis

Data Exploration

- Training data
- Sample from training data

id tweet subtask_a subtask_b subtask_c

73518 @USER He is 😭😭😭 he is so precious 💖 NOT NULL NULL

82921 @USER And why report this garbage. We don't give a crap. OFF TIN OTH

1) DESCRIPTION

The file offenseval-training-v1.tsv contains 13,240 annotated tweets.

The dataset was annotated using crowdsourcing. The gold labels were assigned taking the agreement of three annotators into consideration. No correction has been carried out on the crowdsourcing annotations.

The file offenseval-annotation.txt contains a short summary of the annotation guidelines.

Twitter user mentions were substituted by @USER and URLs have been substitute by URL.

Each instance contains up to 3 labels each corresponding to one of the following sub-tasks:

- Sub-task A: Offensive language identification;

- Sub-task B: Automatic categorization of offense types;

- Sub-task C: Offense target identification.

2) FORMAT

Instances are included in TSV format as follows:

ID INSTANCE SUBA SUBB SUBC

Whenever a label is not given, a value NULL is inserted (e.g. INSTANCE NOT NULL NULL)

The column names in the file are the following:

id tweet subtask_a subtask_b subtask_c

The labels used in the annotation are listed below.

3) TASKS AND LABELS

(A) Sub-task A: Offensive language identification

- (NOT) Not Offensive - This post does not contain offense or profanity.

- (OFF) Offensive - This post contains offensive language or a targeted (veiled or direct) offense

In our annotation, we label a post as offensive (OFF) if it contains any form of non-acceptable language (profanity) or a targeted offense, which can be veiled or direct.

(B) Sub-task B: Automatic categorization of offense types

- (TIN) Targeted Insult and Threats - A post containing an insult or threat to an individual, a group, or others (see categories in sub-task C).
- (UNT) Untargeted - A post containing non-targeted profanity and swearing.

Please note that now targeted threats (TTH) have been merged with targeted insults (TIN) and are listed under Targeted Insult and Threats (TIN). The TTH label present in the trial set is not included in this training set and will not be included in the test set.

Posts containing general profanity are not targeted, but they contain non-acceptable language.

(C) Sub-task C: Offense target identification

- (IND) Individual - The target of the offensive post is an individual: a famous person, a named individual or an unnamed person interacting in the conversation.
- (GRP) Group - The target of the offensive post is a group of people considered as a unity due to the same ethnicity, gender or sexual orientation, political affiliation, religious belief, or something else.
- (OTH) Other – The target of the offensive post does not belong to any of the previous two categories (e.g., an organization, a situation, an event, or an issue)

Please note that now organization are listed under Other (OTH). The ORG label present in the trial set is not included in this training set and will not be included in the test set.

Label Combinations

Here are the possible label combinations in the OffensEval annotation.

- NOT NULL NULL
- OFF UNT NULL
- OFF TIN (IND|GRP|OTH)

- Testing data:

1) DESCRIPTION

The file testset-taska.tsv contains 860 unlabeled tweets.

You are required to upload your sub-task A predictions for each of the 860 instances to CodaLab by no later than 17 Jan 2019 (23:59 UTC).

The evaluations of sub-tasks B and C will be carried out later as previously announced.

You will find ALL the necessary information regarding data format, dates, number of submissions, etc. at CodaLab.

2) FORMAT

Instances are included in TSV format as follows:

ID INSTANCE

The column names in the file are the following:

id tweet

3) TASK AND LABELS

(A) Sub-task A: Offensive language identification

- (NOT) Not Offensive - This post does not contain offense or profanity.

- (OFF) Offensive - This post contains offensive language or a targeted (veiled or direct) offense

In our annotation, we label a post as offensive (OFF) if it contains any form of non-acceptable language (profanity) or a targeted offense, which can be veiled or direct.

- Sample from testing data

id tweet

41438 All two of them taste like ass. URL

Exploratory Visualization

Algorithms and Techniques

The classifiers are Logistic Regression, Random Forest, KNN and SVM.

- Logistic Regression

  It is a statistical method for analysing a data set in which there are one or more independent variables that determine an outcome. The outcome is measured with a dichotomous variable (in which there are only two possible outcomes). The goal of logistic regression is to find the best fitting model to describe the relationship between the dichotomous characteristic of interest (dependent variable = response or outcome variable) and a set of independent (predictor or explanatory) variables.

- Random Forest

  Random forests or random decision forests are an ensemble learning method for classification, regression and other tasks, that operate by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees. Random decision forests correct for decision trees' habit of over fitting to their training set.

- KNN

  The k-nearest-neighbors algorithm is a classification algorithm, and it is supervised: it takes a bunch of labelled points and uses them to learn how to label other points. To label a new point, it looks at the labelled points closest to that new point (those are its nearest neighbors), and has those neighbors vote, so whichever label the most of the neighbors have is the label for the new point (the "k" is the number of neighbors it checks).

- SVM

  The best thing about support vector machines is that they rely on boundary cases to build the much needed separating curve. They can handle non linear decision boundaries. Reliance on boundary cases also enables them to handle missing data for "obvious" cases. SVM can handle large feature spaces which makes them one of the favorite algorithms in text analysis which almost always results in huge number of features where logistic regression is not a very good choice. Result of SVMs are not as not as intuitive as decision trees for a layman. With non linear kernels, SVMs can be very costly to train on huge data.

Benchmark

Decision tree builds classification or regression models in the form of a tree structure. It breaks down a data set into smaller and smaller subsets while at the same time an associated decision tree is incrementally developed. The final result is a tree with decision nodes and leaf nodes. A decision node has two or more branches and a leaf node represents a classification or decision. The topmost decision node in a tree which corresponds to the best predictor called root node. Decision trees can handle both categorical and numerical data.

Methodology

Data Preprocessing

The preprocessing done in the preprocessing class in the “offensive” notebook consists of the following steps:

1. The list of the text is randomized
2. The training text are divided into a training set and a validation set.
3. Using combination of preprocessing methods first (tokenization, stopwords removal and lemmatization).

- Tokenization is the method of breaking the text into smaller components (words and sentences.
- *Stopwords* removes stopwords from text (e.g. removes ‘and’, ‘or’, ‘in’...).
- *Lemmatization* usually refers to doing things properly with the use of a vocabulary and morphological analysis of words, normally aiming to remove inflectional endings only and to return the base or dictionary form of a word, which is known as the *lemma* . If confronted with the token *saw*, stemming might return just *s*, whereas lemmatization would attempt to return either *see* or *saw* depending on whether the use of the token was as a verb or a noun.

- Sample from training data
    1. Before preprocessing

       id tweet subtask_a subtask_b subtask_c

       73518 @USER He is 😭😭😭 he is so precious 💔 NOT NULL NULL

    2. After tokenization, stopwords removal and lemmatization

       Data = `[['precious']]`

       Labels = [0]

    4. Then using vectorization (count).

        - Count

          CountVectorizer to learn the vocabulary of a set of texts and then transform them into a dataframe that can be used for building models.

Implementation

The implementation process can be split into two main stages:

1. The classifier training process
2. Testing stage

During the first stage, the classifier was trained on the preprocessed training data. This is done in Colab notebook (titled “offensive”), and can be further divided into the following steps:

1. Load the training and validation texts into memory, preprocessing them as described in the previous section.
2. Define classifiers and training parameters.
3. Train the classifiers.
4. Calculate confusion matrix, precision, recall and f1 score on testing data.

Results

Model Evaluation

classifier	RandomForest	LogisticRegression	KNN	Benchmark Decision Tree	SVC
precision	0.7526369046758739	0.7505998175172244	0.7348871132937774	0.7089049549637411	0.7754558356860746
recall	0.7575528700906344	0.7573011077542 8	0.7041792547834844	0.7162638469284995	0.6593655589123867
F1 score	0.7412208559 20739	0.7441447222333138	0.6334275176730388	0.7111974573947104	0.5245343167 80419

We can see the Logistic regression and Random Forest beats the benchmark model.

References

- https://medium.com/@sifium/machine-learning-types-of-classification-9497bd4f2e14
- https://www.edvancer.in/logistic-regression-vs-decision-trees-vs-svm-part2/