FRI Natural language processing course 2021
Project report

University of Ljubljana
Faculty of Computer and Information Science

Cross-lingual offensive language identification

Gojko Hajduković

Abstract

To be added.

Keywords

Text classification, Hate speech classification

Advisors: Slavko Žitnik

Introduction

Text classification is one of the most important and well researched tasks in Natural Language Processing. It represents a supervised learning approach whose goal is to classify an input text into the correct category, therefore each input example gets assigned a label and as such is used in wide range of applications. Applications that exploit text classification the most are document classification, web search, information retrieval, sentiment analysis and spam detection. Text classification breaks down to different levels of scope:

1. **`Document level`** - In the document classification algorithm, classifies document as a whole into the categories
2. **`Paragraph level`** - Classifies a portion of a document, a single paragraph into the required categories.
3. **`Sentence level`** - Obtains the categories of a portion of a paragraph, a single sentence.

In this paper we focus on online hate speech classification problem. Online hate speech classification has become a well researched part of NLP and an urgent need since human supervision is unable to handle an enlarging volume of online content. We further define a problem as classifying input text at a paragraph level into binary categories **`1 - Hate speech`** and **`0 - Non hate speech`**. In this paper we will explore two types of models for binary classification problems. First, traditional machine learning models such as `Logistic regression`, `SVM`, `Random forest` and an `Ensemble voting model` are to be employed into classifying online hate speech. The later approach will rely on `BERT` an transformer based machine learning technique. In case of successfully validating our initial hypothesis of correctly classifying hate speech we will introduce a Slovene labeled hate speech data set consisting of 200 comments. Additionally, we will transfer the knowledge from pre-trained multi-lingual models using one-shot transfer and evaluate introduced Slovene data set.

Related work

TO BE ADDED

Methods

In this section we introduce data sets used for training and testing, preprocessing steps as well as all the performed experiments.

Data

For the hate speech classification task we have used five publicly available data sets that have been further processed and adopted in order to get an uniform structured data sets. All five datasets are transformed into a csv file with **`Text`** and **`Label`** columns where label represents binary classification categories. Hate speech is annotated with 1 while non-hate speech is annotated with 0.

Dataset 1

First dataset represents a 1528 Fox news comments [1] which are manually classified into hate and non-hate categories. publicly available at url. The original data set is a json file where each comment has an additional metadata and previous contextual comments. All the metadata have been removed, preserving only the text and the binary label. Table [1] shows that original dataset is highly imbalanced consisting of 435 hateful and 1093 non-hateful comments.

Dataset 2

Second dataset represents a 26581 comments [2] from Reddit posts which are manually classified into hate and non-hate categories, publicly available at url.The original data set is a csv file where each comment has an additional metadata and previous contextual comments. All the metadata have been removed, preserving only the text and the binary label. Table [1] shows that original dataset is highly imbalanced consisting of 5225 hateful and 21356 non-hateful comments .

Dataset 3

Third dataset represents a 38018 comments [2] from Gab posts which are manually classified into hate and non-hate categories, publicly available at url.The original data set is a csv file where each comment has an additional metadata and previous contextual comments. All the metadata have been removed, preserving only the text and the binary label. Dataset consists of 14139 hateful comments and 23888 non-hateful comments.

Dataset 4

Fourth dataset represents a 24802 comments [3] from Twitter posts which are manually classified into hate, offensive and neither categories publicly available at url. Offensive comments have been filtered out from the original dataset resulting into a csv shaped dataset consisting of 1430 hateful comments and 4163 non-hateful ones.

Dataset 5

Fourth dataset represents a 10945 comments [4] from a white supremacists forum which are manually classified into hate and non-hate categories publicly available at url. The original dataset consists of textual documents which hold each comment and a csv file marking whether a document is classified as a hateful or non-hateful one. The dataset has been processed into one csv shaped file consisting of comments and label categories. Table [1] shows that the dataset is highly imbalanced consisting of 1196 hateful comments and 9748 non-hateful comments.

Preprocessing

Preprocessing is applied to all the instances of each dataset that the classifier could predict class for. Preprocessing is conducted through eight steps. Since our datasets are collected mainly from the social media or news media comments there are high number of occurences of emoticons. With step 1 all the emoticons are removed. Same reasoning is behind step 2 and 3 where all twitter urls, hashtags and @name mentions are removed. Steps 4 and 5 remove all non-alphabetical characters and lower down all words. Step 6 tokenizes each input comment with nltk function word_tokenize which firsts tokenizes text into sentences than it further tokeniizes it into tokens. Lemmatization is performed in step 7 in order to get word in a canonical form and get higher frequency of similar words. Nltk WordNetLemmatizer is used for converting a word into lemmas. Additionaly, step 8 combines english stop words removal as well as filtering out words of length 1.

Table 1. Dataset class label distribution.

Imbalanced					
DS 1	DS 2	DS 3	DS 4	DS 5	Class
435	5225	14139	1430	1196	1 - Hate
1093	21356	23888	4163	9748	0 - Regular

Features

For the feature extraction in our baseline model we have used sparse-word embeddings. First, we have used simple frequency count to create feature sets. In the later approach we employed tf-idf technique. With both approaches we have used unigrams and bigrams while using Sklearn implementations of **CountVectorizer** and **TfidfVectorizer** respectively.

Models

Table [1] shows that the data sets are highly imbalanced which can lead to classification algorithms having low accuracy towards the minority class, in our case **hate class**. In order to tackle the problem of imbalanced datasets we have tested under-sampling and over-sampling on training data as well as adjusting class weights.

Traditional ML models

For the baseline models we have selected SKlearn implementation of Logistic regression, Support-Vector Machine and and Ensemble voting classifier - XGBoost.

Logistic regression - We have used sklearn implementation of Logistic Regression classifier. For the model parameter tuning the GridSearchCV is used to choose the best model for predicting classes of each dataset as well as predicting classes for a dataset that is a concatenation of all datasets. Table [] shows the best choice of parameters used for predicting classes.

Support-Vector Machine - We have used sklearn implementation of Support-Vector Machine classifier. For the model parameter tuning the GridSearchCV is used to choose the best model for predicting classes of each dataset as well as predicting classes for a dataset that is a concatenation of all datasets. Table [] shows the best choice of parameters used for predicting classes.

Ensemble models

XGBoost - We have used XGBoost ensemble model for classifying comments data. The boosting algorithm is used with a learning_rate = 0.05, initial n_estimators=300.

Table 2. Model classification weighted results

Imbalanced						
DS 1	DS 2	DS 3	DS 4	DS 5	DS 1-5	Model
0.739	0.845	0.792	0.934	0.865	0.829	LR
0.774	0.865	0.823	0.912	0.901	0.845	SVM

Results

All the models are trained and tested with KFold validation while having **K = 10** with both feature sets created from bag-of-words or tfidf. In the first pass all the models are evaluated on each dataset while in the second pass the models are trained and tested on a dataset which combines all 5 datasets. Since, majority of our datasets are highly imbalanced we have evaluated our models with weighted f1 score as well as with weighted accuracy. Table [3] shows f1 score accross best performing models on all datasets.

References

[1] L. Gao and R. Huang. Detecting online hate speech using context aware models. *ArXiv*, 2018.

[2] Bethke A. Belding E. Qian, J. and W. Yang Wang. A benchmark dataset for learning to intervene in online hate speech. *ArXiv*, 2019.

[3] Automated hate speech detection and the problem of offensive language, year = 2017, journal = ArXiv, author = Davidson, T., Warmsley, D., Macy, M. and Weber, I,.

[4] Ona de Gibert, Naiara Perez, Aitor García-Pablos, and Montse Cuadros. Hate Speech Dataset from a White Supremacy Forum. In *Proceedings of the 2nd Workshop on Abusive Language Online (ALW2)*, pages 11–20, Brussels, Belgium, October 2018. Association for Computational Linguistics.