Nikolov-Radivchev at SemEval-2019 Task 6: Offensive Tweet Classification with BERT and Ensembles

Victor Radivchev
Sofia University, FMI
victor.radivchev@gmail.com

Alex Nikolov
Sofia University, FMI
alexnickolow@gmail.com

Abstract

This paper examines different approaches and models towards offensive tweet classification which were used as a part of the OffensEval 2019 competition. It reviews Tweet preprocessing, techniques for overcoming unbalanced class distribution in the provided test data, and comparison of multiple attempted machine learning models.

1 Introduction

The purpose of this paper is to explore different approaches towards classifying tweets based on whether they are offensive or not, whether offensive tweets are targeted, and identifying the target group of offensive tweets either an individual, a group, or other. Those are the terms of the OffensEval 2019 competition in which we participated. Each of the described activities constituted a separate subtask from the competition. A maximum of three submissions were allowed per subtask which required careful preliminary analysis of the model results during the training phase. A training set of over 13,000 tweets, containing labels for all three subtasks. Each of the subtasks was scored using macro F1 score.

2 Related Work

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 recently as evidenced in publications from the last two years.

Survey papers describing key areas that have been explored for this task include (Schmidt and Wiegand, 2017), (Fortuna and Nunes, 2018) and (Malmasi and Zampieri, 2017). The dataset for this competition is explained in (Zampieri et al., 2019a) and different approaches to the same problem are reported in (Zampieri et al., 2019b).

In order to classify correctly abusive language it is important to analyze its types. A proposal of typology of abusive language sub-tasks is presented in (Waseem et al., 2017) and (ElSherief et al., 2018) examines the target of the speech: either directed towards a specific person or entity, or generalized towards a group of people sharing a common protected characteristic. (Fišer et al., 2017) proposes a legal framework, dataset and annotation schema of socially unacceptable discourse practices on social networking platforms in Slovenia. Finally, a recent discussion on identifying profanity vs. hate speech is presented in (Malmasi and Zampieri, 2018). This work highlighted the challenges of distinguishing between profanity, and threatening language which may not actually contain profane language.

Approaches to detecting hate speech on Twitter using convolutional neural networks and convolution-GRU based deep neural network are discussed in (Gambäck and Sikdar, 2017) and (Zhang et al., 2018) respectively.

Additional related work is presented in workshops such as TA-COS[1], Abusive Language Online[2], and TRAC[3] and related shared tasks such as GermEval (Wiegand et al., 2018) and TRAC (Kumar et al., 2018).

[1] http://ta-cos.org/

[2] https://sites.google.com/site/abusivelanguageworkshop2017/

[3] https://sites.google.com/view/trac1/home

Proceedings of the 13th International Workshop on Semantic Evaluation (SemEval-2019), pages 691–695
Minneapolis, Minnesota, USA, June 6–7, 2019. ©2019 Association for Computational Linguistics

3 Methodology and Data

The data was split into a training and validation set in a ratio of 10:1. All tasks had similar preprocessing and multiple models were trained on the training set. Depending on their performance on the validation set each time the best 3 were submitted.

3.1 Preprocessing

We started our tweet preprocessing by removing most punctuation marks which do not include any useful information for text classification. The symbols '@' and '#' were excluded from the list due to their specific semantics in tweets. Afterwards the tweets were subjected to tokenization and lowercasing.

All occurrences of tokens beginning with a hashtag were split into the separate words comprising the token, provided that each separate word is uppercased. For example, the token #HelloThere is split into two tokens hello and there.

Afterwards we proceeded with removing a variety of different stop words. When training models for the second and third subtask, we excluded personal and possessive pronouns from the list of stop words, as they can contain valuable information for classifying a tweet as targeted or not, or identifying the target group of a targeted tweet. We also attempted lemmatization and spell correction but the results were slightly worse or on par with the ones achieved without using these two techniques.

Pre-trained word vectors on Twitter from project GloVe (Pennington et al., 2014) were used for encoding words to a vector space. Four different vector dimensions were available for use 25, 50, 100, and 200. Although results were slightly better when using higher dimensional vectors, using 200-dimensional vectors proved to have no significant advantage in achieved results over 100-dimensional ones, and proved to be more computationally expensive, which lead us to use 100-dimensional vectors for each subtask.

3.2 Models

We trained a large variety of different models and combined the best of them in ensembles. For all models the embedding layer was freezed, becaused that proved less prone to overfitting.

- Standard Nave Bayes and Support Vector Machine (SVM) from scikit-learn library in python.
- Convolutional Neural Network (CNN) with GlobalMaxPooling and hidden dense layer on top.
- Multilayer Perceptron Network (MLP) with two hidden layers.
- FastText models with n-grams of size 2.
- Recurrent Neural Network (RNN) with GRU units and attention layer and hidden dense layer on top.
- Deep Pyramid Convolutional Neural Network (DPCNN) (Johnson and Zhang, 2017).
- Bidirectional Encoder Representations from Transformers (BERT) (Devlin et al., 2018).
- Soft Voting Classifier (SVC) - averages the predictions of the single models.
- Logistic Regression - meta model trained on half of the validation set with predictions from the single classifiers as features.

3.3 Class imbalance

One of the challenges of the competition was the imbalance of classes for the second and third subtask. We experimented with different techniques for overcoming this challenge:

- Oversampling duplicating some of the examples from the poorly represented classes.
- Class weights assigning lower weights to examples from classes which are better represented and higher weights to examples from classes with a lower overall count.
- Modification of the thresholds used for classifying an example. For example, for a standard binary classification a threshold of 0.5 is applied to the predicted probability in order

to distinguish between the two classes. We attempted to lower this threshold to different levels.

For all model apart from BERT the class weight option was chosen. Only for BERT on subtask C the thresholds were changed instead. For classes OTH and GRP we used thresholds of 0.2 and 0.3 respectively and if any of them was exceeded we would directly assign that class. If both were exceeded we would assign OTH as the class. The coefficients were derived via cross-validation.

4 Results

The results from the test sets for each subtask are displayed below. We have also provided the results from our validation sets, those were the basis upon which we decided which models predictions to submit.

The individual model with the best performance on subtask A was BERT-Large, Uncased with a macro F1 score of 0.781 on the validation set and was selected as one of the models for submission. The other two submitted models were the soft voting classifier with score of 0.788 and the logistic regression model 0.800. The scores of the other trained models are displayed below.

System	F1 (macro)
Logistic Regression	**0.800**
SVC	0.788
BERT-Large	0.781
RNN	0.773
DPCNN	0.768
CNN	0.765
FastText	0.759
Nave Bayes	0.744
MLP	0.742
SVM	0.705

Table 1: Results on the validation set for Sub-task A.

The ensemble models proved to have overfit on the training data and out of the models we have submitted BERT had the highest score, ranking second overall amongst all participants.

In subtask B the highest scoring models on the validation set was the soft voting classifier with a score of 0.64, closely followed by RNN and CNN 0.63. BERT-Base, Uncased performed surprisingly poorly and achieved a score of 0.59.

The soft voting classifier scored the highest on the test set and ranked 16th overall.

System	F1 (macro)	Accuracy
All NOT baseline	0.4189	0.7209
All OFF baseline	0.2182	0.2790
SVC	0.7252	0.8105
Logistic Regression	0.7867	0.8453
BERT-Large	**0.8153**	**0.8547**

Table 2: Results on the test set for Sub-task A.

System	F1 (macro)
SVC	**0.642**
RNN	0.633
CNN	0.631
DPCNN	0.630
Logistic Regression	0.629
MLP	0.614
FastText	0.612
BERT-Base	0.599
Nave Bayes	0.596
SVM	0.576

Table 3: Results on the validation set for Sub-task B.

In subtask C BERT-Base, Uncased was by far the best individual model, achieving a score of 0.64, surpassing its closest contender (Multi-Layered Perceptron) by approximately 0.045. The third model which we submitted was the soft voting classifier with a score of 0.60.

BERT significantly outperformed every other submitted model, securing us first place in subtask C.

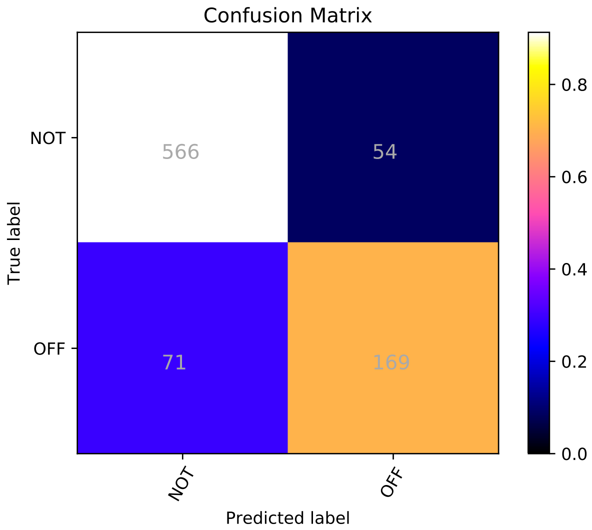

Figure 1: Sub-task A, vradivchev anikolov CodaLab BERT

System	F1 (macro)	Accuracy
All TIN baseline	0.4702	0.8875
All UNT baseline	0.1011	0.1125
RNN	0.6354	0.7667
SVC	**0.6674**	**0.8208**
CNN	0.6248	0.7833

Table 4: Results on the test set for Sub-task B.

System	F1 (macro)
BERT-Base	**0.644**
SVC	0.603
MLP	0.595
Logistic Regression	0.590
RNN	0.586
CNN	0.571
FastText	0.570
DPCNN	0.568
Nave Bayes	0.567
SVM	0.546

Table 5: Results on the validation set for Sub-task C.

System	F1 (macro)	Accuracy
All GRP baseline	0.1787	0.3662
All IND baseline	0.2130	0.4695
All OTH baseline	0.0941	0.1643
BERT-Base	**0.6597**	**0.7277**
MLP	0.5591	0.6808
SVC	0.6107	0.6948

Table 6: Results on the test set for Sub-task C.

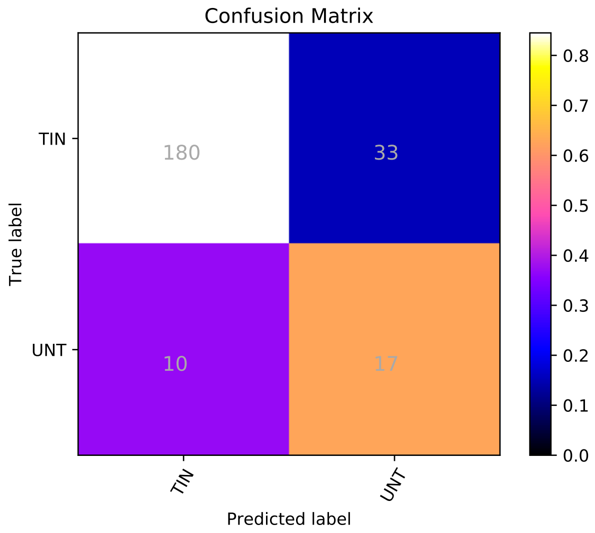

Figure 2: Sub-task B, vradivchev anikolov CodaLab Soft Voting Classifier

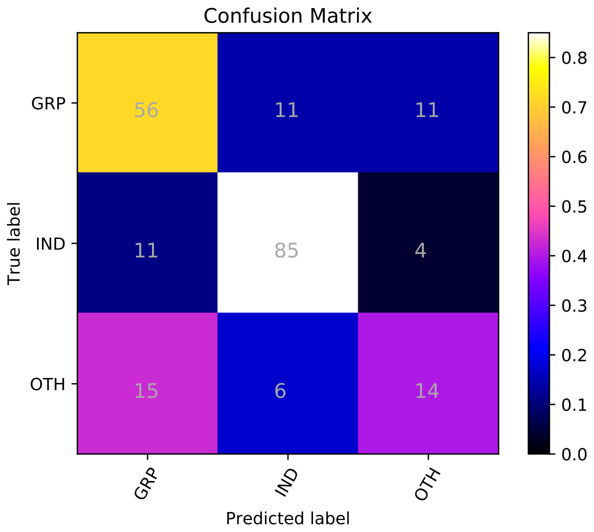

Figure 3: Sub-task C, vradivchev anikolov CodaLab BERT

5 Conclusion

Google's BERT model proved to be a powerful tool for text classification. Not only did it outperform common models on the validation set, but based on the results from the test set it did so without overfitting on the data.

References

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Mai ElSherief, Vivek Kulkarni, Dana Nguyen, William Yang Wang, and Elizabeth Belding. 2018. Hate Lingo: A Target-based Linguistic Analysis of Hate Speech in Social Media. *arXiv preprint arXiv:1804.04257*.

Darja Fišer, Tomaž Erjavec, and Nikola Ljubešić. 2017. Legal Framework, Dataset and Annotation Schema for Socially Unacceptable On-line Discourse Practices in Slovene. In *Proceedings of the Workshop Workshop on Abusive Language Online (ALW)*, Vancouver, Canada.

Paula Fortuna and Sérgio Nunes. 2018. A Survey on Automatic Detection of Hate Speech in Text. *ACM Computing Surveys (CSUR)*, 51(4):85.

Björn Gambäck and Utpal Kumar Sikdar. 2017. Using Convolutional Neural Networks to Classify Hate-speech. In *Proceedings of the First Workshop on Abusive Language Online*, pages 85–90.

Rie Johnson and Tong Zhang. 2017. Deep pyramid convolutional neural networks for text categorization. In *ACL*.

Ritesh Kumar, Atul Kr. Ojha, Shervin Malmasi, and Marcos Zampieri. 2018. Benchmarking Aggression Identification in Social Media. In *Proceedings of the First Workshop on Trolling, Aggression and Cyberbulling (TRAC)*, Santa Fe, USA.

Shervin Malmasi and Marcos Zampieri. 2017. Detecting Hate Speech in Social Media. In *Proceedings of the International Conference Recent Advances in Natural Language Processing (RANLP)*, pages 467–472.

Shervin Malmasi and Marcos Zampieri. 2018. Challenges in Discriminating Profanity from Hate Speech. *Journal of Experimental & Theoretical Artificial Intelligence*, 30:1–16.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *In EMNLP*.

Anna Schmidt and Michael Wiegand. 2017. A Survey on Hate Speech Detection Using Natural Language Processing. In *Proceedings of the Fifth International Workshop on Natural Language Processing for Social Media. Association for Computational Linguistics*, pages 1–10, Valencia, Spain.

Zeerak Waseem, Thomas Davidson, Dana Warmsley, and Ingmar Weber. 2017. Understanding Abuse: A Typology of Abusive Language Detection Subtasks. In *Proceedings of the First Workshop on Abusive Langauge Online*.

Michael Wiegand, Melanie Siegel, and Josef Ruppenhofer. 2018. Overview of the GermEval 2018 Shared Task on the Identification of Offensive Language. In *Proceedings of GermEval*.

Marcos Zampieri, Shervin Malmasi, Preslav Nakov, Sara Rosenthal, Noura Farra, and Ritesh Kumar. 2019a. Predicting the Type and Target of Offensive Posts in Social Media. In *Proceedings of NAACL*.

Marcos Zampieri, Shervin Malmasi, Preslav Nakov, Sara Rosenthal, Noura Farra, and Ritesh Kumar. 2019b. SemEval-2019 Task 6: Identifying and Categorizing Offensive Language in Social Media (OffensEval). In *Proceedings of The 13th International Workshop on Semantic Evaluation (SemEval)*.

Ziqi Zhang, David Robinson, and Jonathan Tepper. 2018. Detecting Hate Speech on Twitter Using a Convolution-GRU Based Deep Neural Network. In *Lecture Notes in Computer Science*. Springer Verlag.