Detecting Offensive Language in Online Texts

Elif Kılık
ID: s0197376
University of Antwerp
elif.kilik@student.uantwerpen.be

Abstract

This paper presents experiments with models trained on the Offensive Language Identification Dataset containing annotated tweets for detecting offensive language. The experiments reproduce learnings from OffensEval 2019 research workshop while focusing solely on task A, which aims to discriminate offensive tweets from non-offensive ones. A state-of-the-art transformer model (BERT: bert-base-uncased) as well as an SVM classifier with a Bag of Words approach are evaluated on additional test sets, including texts from various online sources. Results indicate deep learning techniques outperform the SVM classifier approach, which is in line with the previous research into offensive language detection.

1 Introduction

The augmentation of offensive language use in social media is attracting increasing media as well as public attention. Just recently, platforms such as Twitter has been in the focus of media attention about tackling the issues of hate speech and offensive language (Davies, 2022; Satariano, 2022). Not surprisingly, offensive language detection in text is receiving increasing scholarly attention, especially Natural Language Processing (NLP) community (Markov and Daelemans, 2021), as computational models became more and more capable of automating the intensive process of detecting offensive language in great amounts of text (Zampieri et al., 2019b).

The OffensEval 2019 competition received results from experiments focusing on offensive language detection from 115 competing groups (Zampieri et al., 2019b). The task was divided into three sub-tasks, which were: the sub-task A, classifying tweets as offensive and not offensive, the sub-task B, which categorizes the offensive tweet as targeted or not-targeted and finally the sub-task C, which is about labeling the targeted offensive tweet based on the target (i.e. individual, group, other). 104 competing models were submitted to label tweets as offensive or not-offensive, which is the main focus of the experiments in this paper. The Offensive Language Identification Dataset (OLID) were created for offensive language detection task (Zampieri et al., 2019a; Zampieri et al., 2019b) which incorporates the hierarchical task structure of the competition. To complete the task, various experiments were conducted employing the current state-of-the-art models for offensive language detection on the OLID training set. Afterwards, the models were evaluated with regards to their performance and applicability using the OLID test set, as well as other, out-domain test sets.

2 Methodology

The experiments handle the offensive language identification task as a classification problem, by labelling the texts as offensive vs. not offensive. Differences in pre-processing techniques can result in different model performance (Zampieri et al., 2019b), which points to the importance of pre-processing in order to achieve accurate classification results (Caselli et al., 2020). Below section describes the pre-processing approach of this research.

2.1 Handling Tweets

Twitter language deviates from natural language and tweets can include special forms which can create noise during transformation and vectorization. Mentions to other users and links were already cleaned and replaced with tags '@USER' and 'URL'. Tweets also include hashtags and emojis. During the cleaning process, hashtag signs are removed and the hashtags themselves are represented as words (keywords), and the emojis are converted into word representations using `emot` package for Python. Moreover, repetitive use of letters and punctuations (e.g. 'Hiiii!!!' instead of 'Hi!'), as well as contractions ('yall' instead of 'you all') are corrected. Furthermore, some cases of self-censored profanity were observed in the data. Majority of these cases were caught by human coders and coded as offensive but these cases could be misrepresented during transformation and vectorization. Therefore, the self-censored profanity cases were replaced with the corresponding full words. Finally, the text was cleaned from whitespaces and special characters while punctuations were preserved.

The cleaned tweets are used as input for the training of BERT model without further processing. However, for the SVM classifier, the cleaned tweets are further normalized and tokenized. The English language model of the spacy package, namely `'en_core_web_sm'` is downloaded and applied for lemmatization and POS-tagging of cleaned tweets. Finally, the `TweetTokenizer` from `NLTK` package is used to tokenize tweets, while the additional features are tokenized using `word_tokenize`. [1].

2.2 Models

BERT

The Bidirectional Encoder Representations from Transformers, in short BERT, the pre-trained language model of Google (Devlin et al., 2018) is becoming increasingly popular as it usually over-performs traditional model in natural language tasks. Seven out of top 10 models in OffensEval used BERT to distinguish offensive tweets (Zampieri et al., 2019b). The cleaned versions of tweet texts were used to train various BERT Models which have built-in and trained tokenizer and transformer. For experiments with OLID data, the bert-base-uncased model from Huggingface transformers library was fine-tuned with different parameters and pre-training was applied. The basis for the fine-tuning tests followed the hyper-parameters reported by Caselli et al. (2020) who retrained BERT model for detecting abusive language as well. However, the parameters of the best performing model were set using hyper-parameter tuning techniques as well as numerous trials. The hyper-parameters used for the final model can be found in the appendix. The final model incorporates a maximum sequence length for input, as previous experiments show that limitation of sequence length leads to better performance (Caselli et al., 2020; Zampieri et al., 2019b). The limitation on sequence length was set to be 256, which is higher than the best models of OffensEval but this was done to be able to incorporate the relatively longer texts of Wikipedia test set. Models were run using PyTorch on Google's Colab environment.

SVM Model with Stochastic Gradient Descent Learning

SVM classifiers have been widely used for text classification tasks due to their high performance and achieved high results for offensive text classification (Zampieri et al., 2019a) as well as hate speech detection (Markov and Daelemans, 2021). Experiments with SVM models included the tokenized and lemmatized tweets, as well as the POS-tags of tweets, NRC lexicon emotion associations and insults lexicon of Bassignana et al. (2018) as features, following a similar approach to the work of Markov and Daelemans (2021). To represent tokens as vectors, tfIdf-vectorizer was applied on lemmatized tweets as well as on the above-mentioned additional features. The SVM classifier is optimized with grid search by searching optimal parameters for model loss function, penalty and stopping criteria parameters. Additionally, hyper-parameter optimization is applied on the vectorizer, to find the best n-gram range. Moreover, 5-fold cross validation is applied to analyze error and understand the generalizability of the model.

3 Experiments and Results

3.1 Data

OLID

The OLID dataset contains 14100 tweets which are annotated hierarchically on three levels corresponding to three tasks of OffensEval. The data set is divided into training and test sets, containing 13,240 and 860 tweets respectively.

[1] As opposed to word tokenizer, the Tweet Tokenizer keeps some features belonging to tweets together, making it possible to tokenize @user as one token instead of '@' and 'user'

Additional test data

Testing the model on cross-domain data provides an overview of the general applicability of the model to detect offensive texts. To do this, three additional test sets are obtained, two of which are out-domain sets consisting of online texts, annotated for offensiveness from two sources, namely Reddit and Wikipedia. The former dataset is a sample of Ruddit dataset which includes 1200 Reddit comments in English and the latter dataset is a sample of 1200 Wikipedia posts that are annotated for their offensiveness. The final test set is a data set from Textgain, containing English football tweets which were also annotated as offensive or not offensive. Table 1 reports the distribution of offensive tweets in the training set and test sets.

Data-set Name	**Offensive**	**Not-Offensive**
OLID Training	4400	8840
OLID Test	240	620
Out-Domain: Reddit	543	664
Out-Domain: Wikipedia	600	600
Out-Domain: Textgain	188	1088

Table 1: Distribution of offensive text in different data sets

3.2 Results

The results of the experiments with BERT and SVM model are presented in Table 2. When training BERT, the model selection criterion was set to be macro-averaged F1 score, a harmonic mean of precision and recall, which provides a better overview of model performance compared to accuracy. For a complete comparison, precision, recall and F1-scores (macro-averaged) of the models are reported for in-domain and out-domain test settings as well.

The results of experiments on the OLID test set are in line with the experiments conducted in previous studies (Markov and Daelemans, 2021; Zampieri et al., 2019a; Zampieri et al., 2019b). As expected, the state-of-the-art deep learning model BERT outperformed the SVM classifier. However, the SVM classifier with tf-idf weighted lemma's and the POS-tags of tweets with additional features outperformed the SVM experiments in the OffensEval competition, achieving a macro-averaged F1-score of 0.74 compared to 0.69 (Zampieri et al., 2019b). Performance of the BERT model on OLID test set was comparable to previous studies as well. The resulting macro-average F1-score is the same as the third best result of OffensEval competition (F1: 0.814). Even though similar parameters were used to retrain BERT, the performance with OLID test set was better compared to the findings of Caselli et al. (2020). Markov and Daelemans (2021) achieved even higher performance with their SVM approach as well as their BERT training. These differences in performance underline the importance of decisions regarding pre-processing techniques and feature selection to improve model performance. All in all, it was observed that further pre-training of BERT model on domain specific training data improves the model performance.

SVM

Test Data Name	**Precision**	**Recall**	**F1-score**
OLID Test	0.806	0.718	0.743
Reddit	0.714	0.676	0.673
Wikipedia	0.864	0.863	0.862
Textgain	0.508	0.512	0.500

BERT

Test Data Name	**Precision**	**Recall**	**F1-score**
OLID Test	0.836	0.798	0.814
Reddit	0.719	0.705	0.707
Wikipedia	0.908	0.903	0.903
Textgain	0.546	0.590	0.500

Table 2: Performances of BERT and SVM models tested on in and out-domain test data sets.

While BERT outperforms the SVM approach in general, similar patterns of model performance is observed when comparing across different test sets. Both traditional and deep learning approaches perform better with Wikipedia data compared to social media data, even though the models are trained with social media texts. Similarly, both models performed poorly on Textgain set, even though the source of the data is the same as the training set, i.e. Twitter. These findings mirror the challenges faced by computational approaches to text classification, that are subject to the varieties in language use in different settings, such as different language styles and choice of words per domain and per subject matter.

Language on social media deviates significantly from the structured, grammatically (more) correct language of Wikipedia articles, which could explain why models are able to understand patterns within Wikipedia text and classify these better than texts from social media platforms. On the other hand, the poor performance of both models on Textgain set emphasizes yet another challenge with language analysis, namely domain or topic specific language use through different word choices and structures. One can observe that even if the platform is the same, the attributes and the use of language which make it offensive or not offensive can differentiate a lot per topic and domain. The OLID training set is sampled with social-political keywords to retrieve offensive content on Twitter, while Textgain set used football keywords. Some common aspects of offensive attributes can be observed in political and sports related tweets, such as use of profanity. However, social-political offensive language and offensive language use in comments regarding sports have attributes that are particular to the subject matter. One can expect that these differences are present in tweets as well, which could explain the low performance of SVM and BERT approaches in classifying offensive content in sports context.

Secondly, class imbalance in datasets can create problems in classification and lead to higher rates of false positives and false negatives. While in the OLID sets as well as in the Reddit and Wikipedia sets the offensive texts make up of 30% and 50% of the data sets, the Textgain set contains offensive tweets in 15% of all cases. Computational models tend to favor the frequent class, which could have led models to find patterns of offensiveness in not offensive content and this could explain the high numbers of false positives in Textgain classifications.

4 Conclusion

The aim of this study was to experiment with state-of-the-art methodologies for detecting offensive language. Results are in line with previous research. Findings show that further training of pre-trained language models on domain specific data results in significant improvements in model performance for text classification. Testing models in cross-domain as well as cross-topic settings provide useful insights into generalizability of computational models in different settings. Moreover, evaluating the models on texts from the same social media platform as the training data but on different subject matter revealed that platform specific training should incorporate different subject matters to be able to generalize on platform-wide language tasks. To detect offensiveness on Twitter, we should incorporate offensive content about different matters discussed on Twitter.

References

Elisa Bassignana, Valerio Basile, and Viviana Patti. 2018. Hurtlex: A multilingual lexicon of words to hurt. In *Proceedings of the 5th Italian Conference on Computational Linguistics*, pages 1–6.

Tommaso Caselli, Valerio Basile, Jelena Mitrovic, and Michael Granitzer. 2020. Hatebert: Retraining BERT for abusive language detection in english. *CoRR*, abs/2010.12472.

Pascale Davies. 2022. Will elon musk's twitter become a beacon of free speech or a soap box for hate speech? *Euronews*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.

Ilia Markov and Walter Daelemans. 2021. Improving cross-domain hate speech detection by reducing the false positive rate. In *Proceedings of the Fourth Workshop on NLP for Internet Freedom: Censorship, Disinformation, and Propaganda*, pages 17–22, Online, June. Association for Computational Linguistics.

Adam Satariano. 2022. Musk says his twitter takeover is 'on hold,' then says he's 'still committed'. *The New York Times*.

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. *CoRR*, abs/1902.09666.

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*, pages 75–86, Minneapolis, Minnesota, USA, June. Association for Computational Linguistics.

A Appendix - Hyper-parameters for BERT Model

Hyper-parameters	**Value**
training batch size	32
learning rate	1e-6
warmup steps	0
training epochs	15
adam epsilon	1e-8
max. sequence length	256

Table 3: Hyper-parameters for BERT Model