How Offensive is Negative Sentiment? Using Offensive Language Detection Models for Sentiment Classification

Katarina Ćavar, Sanja Deur, Dorotea Protrka

University of Zagreb, Faculty of Electrical Engineering and Computing
Unska 3, 10000 Zagreb, Croatia
{katarina.cavar, sanja.deur, dorotea.protrka}@fer.hr

Abstract

Removing offensive content on social media was until recently a task that had to be done manually. With the rise of machine learning and natural language processing techniques, automated offensive language identification has become an active area of research. This paper explores a wide range of architectures of both shallow machine learning and deep learning models to accomplish this task. Since an offensive statement can often be viewed as a statement with a negative sentiment, the connection between the tasks of offensive language detection and sentiment analysis is investigated. For this purpose, the success of our models, pretrained on offensive language detection task, is evaluated for the sentiment analysis task and the transferability of models across these two domains is discussed.

1. Introduction

Social media platforms, such as Twitter and Facebook, are a popular way for people to express their feelings and opinions on different matters. The problem arises when those posts start containing hate speech towards another person or a specific group of people, as well as abusive and offensive language. In recent years, there have been various attempts and proposals on how to address this matter, such as paraphrasing the offensive part of a sentence whilst keeping the overall sentiment, as proposed by (dos Santos et al., 2018), or complete removal of some posts, depending on the degree of profanity.

First step in tackling offensive language across social media most certainly is its detection. Considering the enormous amount of data retrieved from social media platforms, desirable solution would include automatic identification of profane language, without or with small amount of manual filtering, as it is very time-consuming and may even cause post-traumatic stress disorder-like symptoms to human annotators (Zampieri et al., 2019a).

Recently, a lot of research has been done in this area and various exceptional models, such as machine learning and deep models, have been developed. Aim of this paper is not on developing a state-of-the-art model, but rather we pursue the construction of different models and comparison of their performance. Our starting point is the OffensEval-2020 shared task which provides the Offensive Language Identification dataset (OLID) with over a 14,000 English tweets (Zampieri et al., 2019b).

Connection between offensive language detection and sentiment analysis is undeniable, due to the fact that offensive language, in most cases, has a negative sentiment. The main difference is the fact that profane language could be used in tweets with positive sentiment to express emphasis (e.g. tweet ”ON HALLOWEEN its aboout to get sooooo F*CKING spooky”). There are mentions in (Zampieri et al., 2019b) of successfully adapting BiLSTM model for offense detection task from a pre-existing model for sentiment analysis.

The second experiment that we have conducted concerns measuring the performance of pretrained models which detect offensive language on the binary Twitter Sentiment Analysis (TSA) problem, through which we attempt to infer the connection and similarity between the two mentioned tasks. To the best knowledge of authors, this idea has not yet been examined by others.

The remainder of this paper is organized as follows: Section 2. presents the overview of prior work done in the field of offensive language identification. Section 3. includes a brief description of datasets and pre-processing techniques used in this work. Section 4. thoroughly describes constructed models, divided into three main categories: (1) baseline model, (2) shallow models and (3) deep models. In section 5. results are provided and commented. Section 6. gives concluding remarks and directions for future work.

2. Related Work

Over the past few years, the field of offensive language detection has arisen a lot of interest in scientific community. One of the latest work is the OffensEval-2020 shared task, the succesor of SemEval-2019 task (Zampieri et al., 2019b) which has greatly influenced our work. For the purpose of offensive language identification, authors usually decide between either using shallow machine learning models, deep learning approach or an ensemble of more methods.

On the one hand, different sorts of machine learning approaches are examined in (Davidson et al., 2017), and the authors finally decide upon using logistic regression with L2 regularization. Furthermore, (Nobata et al., 2016) argue that shallow machine learning based methods outperform deep learning ones, and they manage to outperform a state-of-the-art deep learning approach.

On the other hand, in recent years, deep learning models are gaining popularity and are shown as more successful ones. One approach in detection of offensive language proposed in (Pitsilis et al., 2018) is using ensemble of Long Short Term Memory (LSTM), a powerful type of Recurrent Neural Network (RNN), classifiers. Second interesting approach is a sequentially combined BiLSTM-CNN neural network combined with transfer learning (Wiedemann et al., 2018). The data used in the experiments are more than 5,000 German tweets composed for GermEval-2018 shared task, described in detail in (Wiegand et al., 2018). Lastly, the best solution regarding sub-task A of OffensEval-2020 shared task, (Liu et al., 2019), experimented with linear models, LSTM, and pretrained BERT with finetuning on the OLID dataset.

3. Dataset

3.1. OLID Dataset

Offensive Language Identification dataset (OLID) consists of 14,100 annotated tweets in English. Labels provided for this dataset are divided into three categories based on the sub-task:

- Sub-task A: Offensive language detection
- Sub-task B: Categorization of offensive language types
- Sub-task C: Offensive language target identification

Since our interest lies in offensive language identification, only the first group of labels is used. Tweets are partitioned in two groups: offensive (OFF) and not offensive (NOT). Distribution of tweets is shown in Table 1.

Table 1: Distribution of OLID data

Label	Training	Test	Total
OFF	4,400	240	4,640
NOT	8,840	620	9,460
Total	13,240	860	14,100

3.2. MTSA Dataset

McGill Twitter Sentiment Analysis dataset (MTSA) (Kenyon-Dean et al., 2018) contains over 7,000 tweets across five different topic-domains, annotated by at least five annotators. Tweets are categorized according to following four labels: objective, positive, negative and complicated. Label objective is used in case tweet does not express sentiment, and if it does, sentiment is assigned to one of the remaining three labels. Whilst labels positive and negative are self-explanatory, label complicated is applied when expressed sentiment is ambiguous or mixed. Custom-made distribution of data is shown in Table 2 and one can observe that, even though there are half as many tweets, the ratios are similar as in the Table 1.

For the purposes of this paper, labels are condensed into two labels: negative (NEG) and non-negative (NOT). Since this dataset originally explores the problem of complicated tweets, and we wanted to focus strictly on the ability of our models to be applied to a different domain problem, we've taken into account only those tweets which had over 80% consensus of the annotators. Also to have a balanced dataset for testing, only a portion of that remaining dataset was chosen in order to have the same number of instances for each class. More specifically, the final test set contained 600 tweets labeled as NEG and 600 labeled as NOT.

Table 2: Distribution of MTSA data

Label	Training	Test	Total
NEG	2,213	147	2,360
NOT	4,437	316	4,753
Total	6,650	463	7,113

3.3. Pre-processing

For pre-processing the tweets several techniques are applied, such as normalizing the hashtags, mentions and elongated words, removing stop words and punctuation, converting words to lowercase etc. Described goal is achieved using standard tokenizers (Standard Core NLP), such as WordPunctTokenizer and RegexpTokenizer, as well as the NLTK TweetTokenizer [1] and TextBlob [2].

4. Models

4.1. Baseline Model

Simple baseline model goes through all of the words in a tweet and checks whether they appear in the dictionary of offensive words, which is constructed from three publicly available datasets[3]. Best results are attained whilst using the NLTK TweetTokenizer for pre-processing purposes, with accuracies being 0.705 and 0.701 for OLID and MTSA dataset, respectively. Interestingly enough, accuracy scores are similar for both offense and sentiment tasks, which indicates that tweets with negative sentiment oftentimes contain offensive words. This finding gave us the reassurance to pursue the testing on further, nontrivial models, and perform a comparison between the two tasks.

4.2. Shallow Machine Learning Models

The shallow model used in this paper is logistic regression, with different features explored. For the first variant of logistic regression, features are extracted manually and are focused on negative connotation of words in a tweet or tweet itself. Ablation study was performed in order to decide which features should be used. Selected features are:

- Negative sentiment polarity - a single number extracted using VADER (Hutto and Gilbert, 2015) sentiment analysis system;
- Negative words - boolean feature indicating presence of words from the list of negative words (Hu and Liu (2004), Liu et al. (2005));

[1] http://www.nltk.org/api/nltk.tokenize.html

[2] https://textblob.readthedocs.io/en/dev/

[3] https://www.kaggle.com/nicapotato/bad-bad-words
https://gist.github.com/jamiew/1112488
https://github.com/LDNOOBW/List-of-Dirty-Naughty-Obscene-and-Otherwise-Bad-Words/blob/master/en

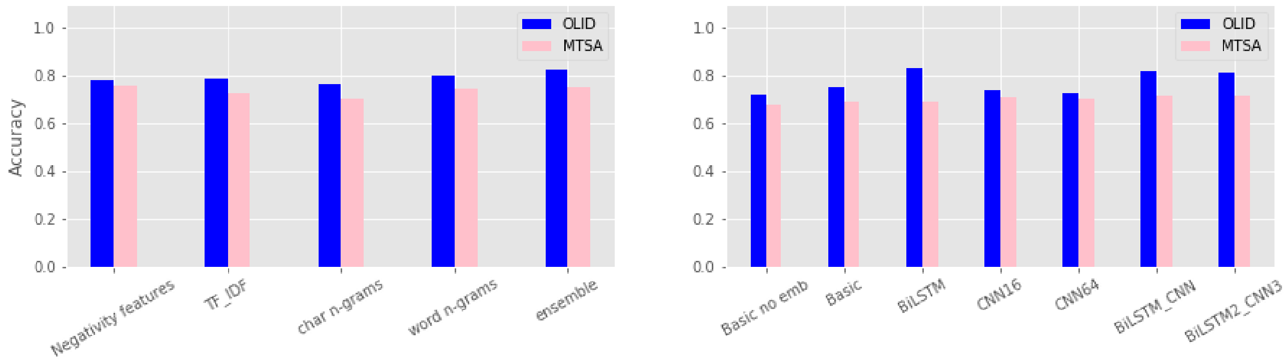

Figure 1: Comparison of different models for offensive language detection over 2 datasets: OLID and MTSA

Table 3: Performance of shallow machine learning models

Model	Accuracy	Precision	Recall	F1
LR - negativity features	0.779	0.711	0.358	0.476
LR - TF-IDF unigrams	0.791	**0.866**	0.296	0.441
LR - char n-grams	0.764	0.599	**0.491**	0.540
LR - word n-grams	0.798	0.770	0.404	0.530
Ensemble	**0.823**	0.848	0.467	**0.602**

- Bad/offensive words - boolean feature indicating presence of words from the list of bad words[4].

Second variant contains unigram features weighted by the term frequency – inverse document frequency (TF-IDF). Last two variants use n-grams of different sizes weighted by the token counts as features. Second variant's features are character n-grams of sizes from 3 to 5 and third's are word n-grams of sizes 1, 2 and 3.

Finally, ensemble is built based on majority voting, whilst using only the hard predictions. Since there is an even number of models, both labels could get the same number of votes. To solve this problem, the predictions of the model which on average performed the best were taken into account twice. Only time when voting is not based on the majority is in the case when none of the words from a tweet have been recognized, which, most of the times, indicates that words have been misspelled (due to slang or human error). Therefore, in such occasions, the prediction of character-based variant is the only one used, which has been proven to perform well.

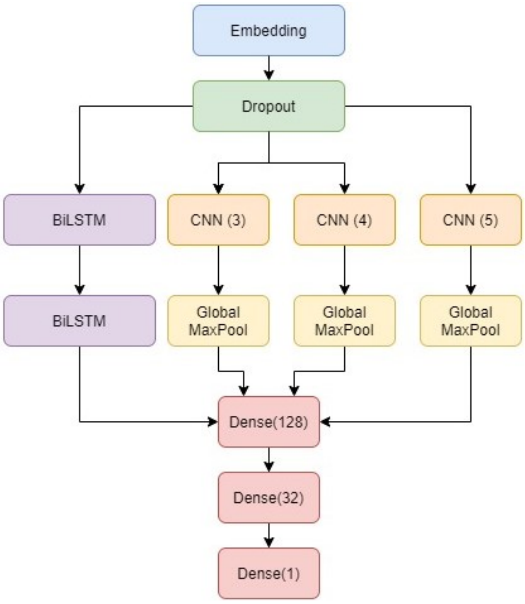

Figure 2: BiLSTM2_CNN3 model architecture

4.3. Deep Learning Models

For deep learning approach, we have explored various architectures. As the baseline, two simple dense-only models have been developed - BasicNoEmb, without pretrained embeddings, and Basic, with pretrained GloVe 200d pretrained embeddings.

All the other models have also been trained upon pretrained GloVe 300d word embeddings. In the Figure 3. (right), a few of those architectures are shown. Various combinations of Bidirectional LSTMs and CNNs are explored, in some cases used by themselves (but still combined with dense layers in the end), while in other cases LSTMs and CNNs are combined, with the intention of attaining better scores. The models shown in the Table 4 are BiLSTMs which consist of an embedding layer, BiLSTM layer and three dense layers. CNN models have a 1D convolutional layer and

[4] https://www.freewebheaders.com/bad-words-list-and-page-moderation-words-list-for-facebook/

Table 4: Performance of deep learning models

Model	Accuracy	Precision	Recall	F1
BasicNoEmb	0.7223	0.529	0.512	0.502
Basic	0.752	0.564	0.571	0.550
BiLSTM	**0.829**	**0.785**	0.502	0.597
CNN16	0.741	0.545	**0.637**	0.569
CNN64	0.726	0.520	0.625	0.548
BiLSTM_CNN	0.821	0.694	0.612	0.633
BiLSTM2_CNN3	0.815	0.673	**0.637**	**0.644**

a global max pooling layer instead of BiLSTM. In Table 4, two versions can be seen - CNN16 with 16 filters and CNN64 with 64 filters as the output. Both CNN models have kernel size of 3 and a stride of 1. BiLSTM_CNN model has a BiLSTM in parallel with a CNN of kernel 3 and stride 1, whose outputs concatenated together represent the input into 3 dense layers. Architecture of the most complicated model, BiLSTM2_CNN3, is shown in Figure 2. This model consists of four branches whose concatenated outputs go into dense layers. In the first branch, we have two BiLSTM layers on top of each other, and in all the other branches are CNNs with kernel sizes of 3, 4 and 5 respectively.

5. Results

In search for a good model to represent an indicator of offensive language, a lot of models have been subjected to testing. Performance of which we can see in two tables: table 3 provides results of shallow models and table 4 of deep ones.

Using t-test it is observed that with great confidence ($p < 0.001$) it can be concluded that all shallow ML models outperform baseline model. The best shallow model is ensemble of models (t-test, $p < 0.01$) which profits from all virtues of other variants. Because of LR char n-gram, ensemble can notice even misspelled swearword or some other word with negative connotation. It is also interesting to note that model with just 3 features (LR - negativity features) can achieve such results and become comparable with other, more complicated models.

Regarding deep models, it can be observed in the Figure 1 that the models that include BiLSTMs tend to perform a bit better than the ones that don't. It is noticeable that deep models perform differently based on their network structure. However, with 97,5% confidence we can say that all but two basic models outperform baseline. Without an extensive parameter grid search, we've found empirically that having tweets pre-processed helps in the process of learning, that having less CNN filters often provides better or approximately the same score as having more CNN filters and that usually our models struggled more with recall than with precision. But overall all the models that include a BiLSTM perform very similarly.

One of our main questions is if a model trained on the task of classifying offensive language can be used in a domain of classifying sentiment. As explained in Section 3.2, we've transformed the original MTSA dataset into a dataset which has two classes: "negative" and "non negative". Our assumption is that negative tweets will likely contain some of the words and constructions similar to ones contained in an offensive tweet, however sometimes tweets can be negative and non-offensive, so the models pretrained on OLID were expected to perform a bit worse on the MTSA dataset. As shown in Figure 1, it can be observed that indeed the models always perform slightly worse on MTSA than on OLID, but still they perform much better than a dummy classifier which would randomly assign labels to each input, their performance is well over expected 50%. We've explored some tweets for which the classifier was most certain (tweets whose model outputs were either lower than 0.1 which indicates that the model is pretty sure in the label NON, or higher than 0.9 what makes it sure in NEG). One of the expected mistakes were tweets that are negative, but not offensive, like "I had the meanest headache of life yesterday, fell asleep at 5 and just now waking up.. what is life". In contrast, in the example "My mom is going away tonight so imma have a sh*t ton of coffee for dinner" was classified as offensive, and its true label is non-negative. We assume that the model recognized an offensive word ("sh*t") inside these tweets and automatically labeled it as offensive. In conclusion, the results have indicated good performance on this experiment.

6. Conclusion

Nowadays, because of the extensive usage of social media, offensive language is becoming a big problem and detection of such behaviour is not a trivial task. In this paper, we proposed a number of different models for solving the task of offensive language detection. From obtained results, it is obvious that some offensive language can be detected, either by extracting some simple features such as presence of negatively connoted words, or by extracting more sophisticated features using pretrained word embeddings. On the other hand, recall values have shown that a lot of offensive tweets have still passed undetected.

The second goal of this paper was to check if models that classify offensive language can also detect negative sentiment. This experiment was conducted on another dataset (MTSA) with models pretrained on OLID dataset. As expected, the results were slightly worse for sentiment analysis then for offensive language detection but were way better than classification done by random.

References

Thomas Davidson, Dana Warmsley, Michael Macy, and Ingmar Weber. 2017. Automated hate speech detection and the problem of offensive language. In *Eleventh international aaai conference on web and social media*.

Cicero Nogueira dos Santos, Igor Melnyk, and Inkit Padhi. 2018. Fighting offensive language on social media with unsupervised text style transfer. In *56th Annual Meeting of the Association for Computational Linguistics*.

Minqing Hu and Bing Liu. 2004. Mining and summarizing customer reviews. pages 168–177, 08.

C.J. Hutto and Eric Gilbert. 2015. Vader: A parsimonious rule-based model for sentiment analysis of social media text. 01.

Kian Kenyon-Dean, Eisha Ahmed, Scott Fujimoto, Jeremy Georges-Filteau, Christopher Glasz, Barleen Kaur, Auguste Lalande, Shruti Bhanderi, Robert Belfer, Nirmal Kanagasabai, et al. 2018. Sentiment analysis: It's complicated! In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1886–1895.

Bing Liu, Minqing Hu, and Junsheng Cheng. 2005. Opinion observer: Analyzing and comparing opinions on the web. 05.

Ping Liu, Wen Li, and Liang Zou. 2019. Nuli at semeval-2019 task 6: transfer learning for offensive language detection using bidirectional transformers. In *Proceedings of the 13th International Workshop on Semantic Evaluation*, pages 87–91.

Chikashi Nobata, Joel Tetreault, Achint Thomas, Yashar Mehdad, and Yi Chang. 2016. Abusive language detection in online user content. In *Proceedings of the 25th international conference on world wide web*, pages 145–153.

Georgios K Pitsilis, Heri Ramampiaro, and Helge Langseth. 2018. Detecting offensive language in tweets using deep learning. *arXiv preprint arXiv:1801.04433*.

Gregor Wiedemann, Eugen Ruppert, Raghav Jindal, and Chris Biemann. 2018. Transfer learning from lda to bilstm-cnn for offensive language detection in twitter. *arXiv preprint arXiv:1811.02906*.

Michael Wiegand, Melanie Siegel, and Josef Ruppenhofer. 2018. Overview of the germeval 2018 shared task on the identification of offensive language.

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).