nlpUP at SemEval-2019 Task 6: A Deep Neural Language Model for Offensive Language Detection

Jelena Mitrović, Bastian Birkeneder, Michael Granitzer
Faculty of Computer Science and Mathematics
University of Passau, Germany

jelena.mitrovic@uni-passau.de | birkeneder@fim.uni-passau.de
michael.granitzer@uni-passau.de

Abstract

This paper presents our submission for the SemEval shared task 6, sub-task A on the identification of offensive language. Our proposed model, C-BiGRU, combines a Convolutional Neural Network (CNN) with a bidirectional Recurrent Neural Network (RNN). We utilize word2vec to capture the semantic similarities between words. This composition allows us to extract long term dependencies in tweets and distinguish between offensive and non-offensive tweets. In addition, we evaluate our approach on a different dataset and show that our model is capable of detecting online aggressiveness in both English and German tweets. Our model achieved a macro F1-score of 79.40% on the SemEval dataset.

1 Introduction

The ever-increasing amount of user-generated data introduces new challenges in terms of automatic content moderation, especially regarding hate speech and offensive language detection. User content mostly consists of microposts, where the context of a post can be missing or inferred only from current events. The challenge of automatic identification and detection of online aggressiveness has therefore gained increasing popularity in the scientific community over the last years.
Several recent workshops and conferences such as TRAC (Kumar et al., 2018), ALW2 (Fišer et al., 2018), and GermEval (Wiegand et al., 2018) show the growing importance of this subject. The SemEval 2019 shared task 6 (Zampieri et al., 2019b) further addresses this topic by introducing the Offensive Language Identification Dataset (OLID), which consists of tweets, labeled with a three-level annotation model (Zampieri et al., 2019a). Sub-task A is composed of a binary classification problem of whether a tweet in the dataset is offensive or not. Sub-task B focuses on different categories of offensive language and the goal of sub-task C is to identify the targeted individual of an offensive tweet.

In the following paper, we present our contribution to sub-task A. After the related work section, we outline our conducted experiments in section 3 and further describe the used baseline model, as well as the submitted model. In section 4 we report the results of our experiments on the OLID dataset and the additionally used GermEval dataset. Section 5 discusses our results and section 6 concludes our work and describes possible future work.

2 Related Work

Several methods and models have been presented in literature over the last decade to address the predicament of identifying hate speech, offensive language, and online aggressiveness. In the following section, we present the most notable contributions related to our work.
The tweets collected by Davidson et al. (2017) were divided into Hate, Offensive, and Neither. Their proposed algorithm uses unigram, bigram, and trigram tokens as features, weighted by the respective TF-IDF, as well as Part-of-Speech (POS) tagging and different metrics to determine the readability and sentiment of a tweet. Logistic-regression and linear SVM result in the best performance for a wide range of assessed classifiers. Nobata et al. (2016) collected comments from Yahoo! Finance and News articles over a time period of one year and labeled them as either 'Abusive' or 'Clean'. They experimented with various different features, including n-gram, linguistic, syntactic, and distributional semantics features.

Various approaches utilized deep learning models for text categorization. Zhang et al. (2015) proposed a character-level convolutional network for text classification on large-scale datasets. Their network uses 1-dimensional convolutional filters to extract features from different character embed-

dings. Gambäck and Sikdar (2017) further experimented with convolutional networks in the context of online hate speech classification. Their research work compares different types of convolutional models, namely character-level, word vectors with a pretrained word2vec (w2v) model, randomly generated word vectors, and w2v in combination with character n-grams. The results of their experiments suggest that w2v embeddings are the most suitable for this task. Zhang et al. (2018) suggest an architecture similar to our network, where a convolutional filter extracts features from pretrained word embeddings. After max pooling, the feature maps are processed using a unidirectional GRU. Their model is compared to a bag-of-n-gram model on various multi-class hate speech datasets and shows promising results. A detailed survey on different architectures, methods and features for offensive language detection is provided by Schmidt and Wiegand (2017).

3 System Description

In addition to Twitter data provided by the organizers of the SemEval shared task, we further evaluate our approach on German tweets from the GermEval (2018) shared task. The OLID dataset contains 13,240 tweets, with 4,400 offensive and 8,840 non-offensive tweets (66.77% offensive, 33.23% non-offensive). Similarly, the GermEval dataset contains 5,009 tweets, divided into 1,688 offensive and 3,321 non-offensive tweets (66.30% offensive, 33.70% non-offensive). To compensate for the imbalanced class distributions and weigh each class equally, we choose the macro averaged F1-score of both classes as our main evaluation metric. From both data sets we use 10% of our tweets as test set. The remaining tweets are split into 90% training set and 10% validation set. We conduct a stratified 10-fold cross-validation on the training and validation set to prevent overfitting and to validate our model.

The pretrained w2v model, which is used to initialize the weights of our embedding layer, resulted from the work of Godin et al. (2015). The w2v model for the GermEval dataset originates from our previous work (2018).

For comparison to our proposed model, a token bag-of-n-gram model composed of unigrams, bigrams, and trigrams weighted by their TF-IDF is used as baseline approach. We subsequently analyze the performance of different classifiers on the resulting feature space.

We have used the packages *keras*, *scikit-learn*, *gensim*, and *nltk* for preprocessing and the implementation of our models.

3.1 Preprocessing

Tweets are first tokenized and converted to lowercase. We constrain repeated character sequences to length 3 and replace all longer character sequences. HTML character encodings are replaced by their corresponding literal or token representation (e.g. '*&*' translates to '*and*'). Tokens are further split if they enclose a set of special characters ('\', '/', '&', '-'). Since hashtags are often used to replace contextually important words mid-sentence, we split hashtags in the actual hash-symbol and the following string to keep the semantic information of a hashtag (e.g. '*Brainless #Liberal Stooge Ocasio-Cortez*').

3.2 Baseline Model

A TF-IDF bag-of-words model as baseline approach is chosen to evaluate the performance of our model. We limit our feature space to the 10,000 most frequently used unigrams, bigrams, and trigrams in a corpus. Furthermore, we stem each token in the preprocessing phase and remove stopwords. We compare the performance of several classifiers, namely multinomial Naive Bayes (NB), SVM, Decision Tree (DT), and Logistic Regression (LogR) and conduct a grid search to optimize our hyper-parameters.

3.3 C-BiGRU

After the preprocessing step, we construct a dictionary which maps all unique tokens to their number of occurrences in the respective corpus. Tokens which appear only once in a corpus are disregarded and treated as unknown token. As a next step, we construct the weighting matrix $W^{m \times dim}$ for our embedding layer, where dim is the dimension of the used w2v model and m the number of unique tokens $t_i, i \in \{1, ..., m\}$. The word vector of t_i is stored in W if the token is represented in the w2v model. If t_i has no pretrained word vector, we generate a random vector drawn from the uniform distribution within $\left[-\sqrt{\frac{6}{dim}}, \sqrt{\frac{6}{dim}}\right]$ as suggested by He et al. (2015). We fix the maximum length of a sentence to 150 tokens, longer sequences are clipped at the end and shorter sequences are padded with a masking token.

The convolutional layer of our classifier consists of $(k \times 128)$ 1-dimensional filters, where k is the number of different window sizes. These window sizes range from 2 to 5 and allow the extraction of n-gram features. The padding of the input is kept constant, resulting in the same output sequence length as the input. We further choose ReLu as activation function. The resulting feature maps are concatenated and passed towards the recurrent layer.

Gated Recurrent Units (GRU) as initially proposed by Cho et al. (2014) are used in RNNs to capture long-term dependencies of input sequences. Similar to Long Short-Term Memory (LSTM) units (Hochreiter and Schmidhuber, 1997) GRU are able to overcome the vanishing gradient problem by using a gating mechanism. GRU have shown to achieve comparable results to LSTM in sequence modeling tasks and are able to outperform the latter on smaller data sets (Chung et al., 2014). The recurrent layer in our model consists of a bidirectional GRU, where the concatenated feature maps, which resulted from the convolutional layer, are used as input for the GRU layer. Simultaneously, the reversed copy of the input sequence is used for the second GRU layer. Both GRU layers return a hidden state for each processed feature map. The output of both layers is then concatenated. We set the length of the returned hidden states to 64 for both layers, resulting in an output space of (150×128) neurons.

Afterwards, a global max pooling layer reduces the output space to (1×128) nodes. The following fully-connected layer consists of 32 neurons, which connect to a single output neuron. The output neuron utilizes the *sigmoid* activation function.

To additionally prevent overfitting, we include two dropout layers with a dropout rate of 0.2; one after the embedding layer and another one after the fully-connected layer. Furthermore, we adopt early stopping and use 10% of the training data as validation split. We use cross entropy as error function for our model and the optimizer 'adam' to update our network weights (Kingma and Ba, 2014). The batch size for the gradient update is set to 32. A schema of our proposed model is illustrated in Figure 1.

4 Results

For the comparison model, the SVM performs best on the OLID dataset with an F1-score of 70.22% averaged over a 10-fold cross-validation. The SVM also shows the best results on the GermEval dataset with an F1-score of 66.61%. The evaluation on the test set results in 66.78% F1-score for the GermEval gold test set. The evaluation of the baseline model for the OLID gold test set is not possible at the time of writing, since the gold test data have not yet been released.

The C-BiGRU achieved a 76.28% F1-score on the OLID and a 71.13% F1-score on the GermEval dataset on average over a 10-fold cross-validation. On the OLID gold test set, our model achieved an F1-score of 79.40%. The evaluation on the GermEval gold test data resulted in a 72.41% F1-score. An overview of all results can be found in Table 1. Figure 2 shows the confusion matrix of our submitted predictions for the SemEval shared task.

	Baseline		C-BiGRU	
	CV	gold	CV	gold
OLID	70.22%	-	76.28%	79.40%
GermEval	66.61%	66.78%	71.13%	72.41%

Table 1: All results in table form (CV = cross-validation; gold = gold test set).

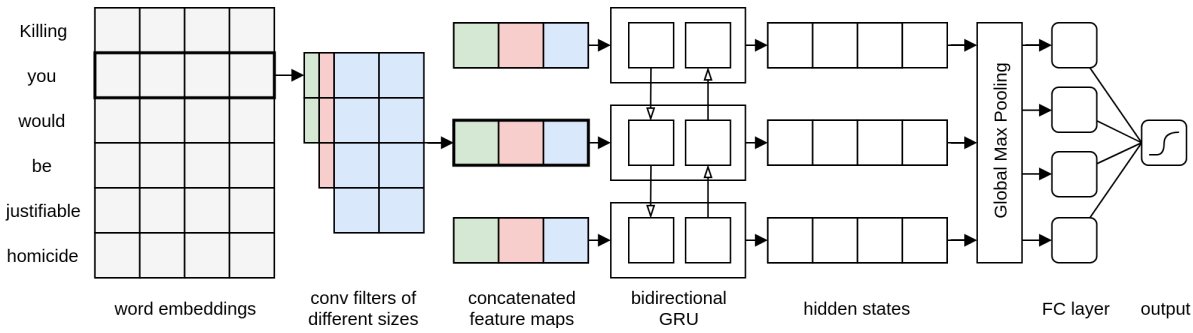

Figure 1: Representation of the proposed classifier.

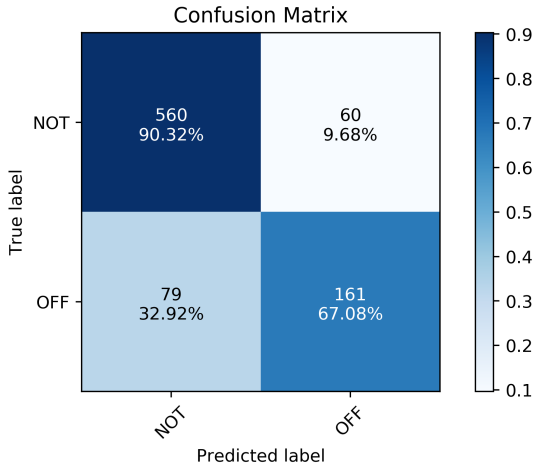

Figure 2: Confusion Matrix of the OLID gold test set, sub-task A. Depicted are instances and normalized values.

5 Discussion

The presented model continues our work on the identification of offensive German tweets (2018). We were able to improve our proposed model by adjusting the architecture of the recurrent layer in our neural network. By using a bidirectional GRU instead of a unidirectional LSTM, we are able to capture past and future information about the input sequence and exploit the better performance of GRU networks on smaller datasets. Furthermore, we return the hidden states for each feature map instead of returning only the last hidden state. This allows us to extract higher-level sequentially dependent features from each concatenated feature map.

Our experiments show that our suggested model outperforms the baseline model on both datasets. The difference between the F1-scores for the English and German dataset might be attributed to the smaller size of the German training set, which contains only about 5,000 tweets. The discrepancy between the results of our cross-validation and achieved score on the OLID test set might be explained by the small amount of test tweets, which may lead to imprecise results for the submitted runs.

By utilizing w2v as features, we are able to limit extensive and language specific preprocessing.

> "@USER Lolol God he is such an a**hole."

In this example, the vector representation of "a**hole" has a high cosine similarity (0.63) to the vector representation of "asshole", which allows our model to classify this tweet as offensive. On the contrary, our approach falls short when confronted with indirect insults.

> "@USER @USER Im sure the air that he is breathing is also bad."

Our model wrongly predicts a non-offensive tweet in this instance.

The detection of offensive, hateful, racist, and/or sexist user behavior in social media still proves to be a challenge. Even for humans, it can be problematic to identify offensive microposts, since these posts can be ambiguous and dependant on the personal mindset of a reader. Ross et al. (2017) show that it can be difficult to measure the agreement of annotators about hate speech in the light of the European refugee crisis. They conclude that instead of a classification problem, a regression model with an average offensiveness score of multiple annotators might be more suitable for this task. Furthermore, it can be difficult to grasp the full context of an arbitrary tweet. With only excerpts of a conversation, the context and true intention of the author may be difficult to determine.

6 Conclusion and Future Work

In this paper, we describe our submitted model for the SemEval shared task 6 and evaluation methods for the identification of online aggressiveness in social media microposts. Our model achieves good results in the two evaluated datasets. For the OLID dataset which contains English tweets, a macro F1-score of 79.40% is reached, while our network resulted in an F1-score of 72.41 % on the GermEval dataset, which consists of German tweets.

We plan to evaluate our approach on more datasets to further investigate the potential of our model for different languages. One such set is the TRAC dataset, which contains aggression-annotated Facebook posts and comments in Hindi. Furthermore, we want to examine whether additional features such as character-level embeddings or POS tagging will improve our results. Inclusion of figurative language detection has proved to enhance many NLP tasks, such as argument mining and so-called hidden hate speech (Mitrović et al., 2017), which is also one of our future directions.

References

Bastian Birkeneder, Jelena Mitrović, Julia Niemeier, Leon Teubert, and Siegfried Handschuh. 2018. upInf - Offensive Language Detection in German Tweets. In *Proceedings of the GermEval 2018 Workshop*, pages 71 – 78.

Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078v3*.

Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*.

Thomas Davidson, Dana Warmsley, Michael Macy, and Ingmar Weber. 2017. Automated hate speech detection and the problem of offensive language. *arXiv preprint arXiv:1703.04009*.

Darja Fišer, Ruihong Huang, Vinodkumar Prabhakaran, Rob Voigt, Zeerak Waseem, and Jacqueline Wernimont. 2018. Proceedings of the 2nd workshop on abusive language online (alw2). In *Proceedings of the 2nd Workshop on Abusive Language Online (ALW2)*.

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.

Fréderic Godin, Baptist Vandersmissen, Wesley De Neve, and Rik Van de Walle. 2015. Multimedia lab @ acl wnut ner shared task: Named entity recognition for twitter microposts using distributed word representations. In *Proceedings of the Workshop on Noisy User-generated Text*, pages 146–153.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2015. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pages 1026–1034.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Ritesh Kumar, Atul Kr. Ojha, Marcos Zampieri, and Shervin Malmasi. 2018. Proceedings of the first workshop on trolling, aggression and cyberbullying (trac-2018). In *Proceedings of the First Workshop on Trolling, Aggression and Cyberbullying (TRAC-2018)*. Association for Computational Linguistics.

Jelena Mitrović, Cliff O'Reilly, Miljana Mladenović, and Siegfried Handschuh. 2017. Ontological representations of rhetorical figures for argument mining. *Argument & Computation*, 8(3):267–287.

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. International World Wide Web Conferences Steering Committee.

Björn Ross, Michael Rist, Guillermo Carbonell, Benjamin Cabrera, Nils Kurowsky, and Michael Wojatzki. 2017. Measuring the reliability of hate speech annotations: The case of the european refugee crisis. *arXiv preprint arXiv:1701.08118*.

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*, pages 1–10.

Michael Wiegand, Melanie Siegel, and Josef Ruppenhofer. 2018. Overview of the germeval 2018 shared task on the identification of offensive language. *Austrian Academy of Sciences, Vienna September 21, 2018*.

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)*.

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. In *Advances in neural information processing systems*, pages 649–657.

Ziqi Zhang, David Robinson, and Jonathan Tepper. 2018. Detecting hate speech on twitter using a convolution-gru based deep neural network. In *European Semantic Web Conference*, pages 745–760. Springer.