# hpiDEDIS at GermEval 2019: Offensive Language Identification using a German BERT model

**Julian Risch**[1] **Anke Stoll**[2] **Marc Ziegele**[3] **Ralf Krestel**[4]

[1]Hasso Plattner Institute, University of Potsdam, julian.risch@hpi.de
[2,3]Heinrich Heine University Düsseldorf, anke.stoll@hhu.de, ziegele@phil.hhu.de
[4]University of Passau, ralf.krestel@uni-passau.de

## Abstract

Pre-training language representations on large text corpora, for example, with BERT, has recently shown to achieve impressive performance at a variety of downstream NLP tasks. So far, applying BERT to offensive language identification for German-language texts failed due to the lack of pre-trained, German-language models. In this paper, we fine-tune a BERT model that was pre-trained on 12 GB of German texts to the task of offensive language identification. This model significantly outperforms our baselines and achieves a macro F1 score of 76% on coarse-grained, 51% on fine-grained, and 73% on implicit/explicit classification. We analyze the strengths and weaknesses of the model and derive promising directions for future work.

## 1 Offensive Language in Online Media

Social media, micro-blogging, and comparable participatory platforms can offer freely accessible discussion spaces and the possibility of communicative integration of different social and interest groups. For this reason, they represent an important cornerstone of modern democracies. In reality, however, online discussions are often the scene of violence, abuse, and incivility (Coe et al., 2014). Studies have shown that offensive and abusive communication makes participants withdraw from online discussions (Springer et al., 2015). Additionally, offensive language can promote aggressive cognitions and negative emotions (Rösner et al., 2016), and reinforce negative prejudices against social groups (Hsueh et al., 2015). The automated detection of offensive language and related concepts, such as incivility, hate speech, or toxicity could help to counter such effects by supporting moderators in effectively identifying and responding to offensive content in online discussions.

This paper presents an approach of detecting different forms of offensive language including profanity, insult, and abuse, and explicit and implicit offensive language in German-language tweets using BERT (Bidirectional Encoder Representations from Transformers). In the GermEval Shared Task 2 (2019), our best systems achieve macro F1 scores of 76.4% on coarse-grained, 51.2% on fine-grained, and 73.1% on implicit/explicit classification.

## 2 Related Work

Offensive language identification and related tasks, such as the detection of toxicity and hate speech, have recently gained popularity within the Natural Language Processing (NLP) community. These tasks are particularly challenging from an NLP perspective, since hate speech, toxicity, or offensive language are often not explicitly communicated through the use of unique offensive words. Further, many words are used with different meanings in different contexts. Traditional lexical and bag-of-words (BoW) approaches often struggle in identifying implicit and context-related forms of offensive language. Davidson et al. (2017) found that only five percent of tweets that contain words of the hate speech lexicon Hatebase.org were flagged as hate speech by human annotators. In their study on anti-black racism on Twitter, Kwok and Wang (2013) show that tweets are classified as abusive based on words such as “black” or “white”, which bear no racist undertones of their own.

Deep Learning (DL) methods marked a significant step forward in the detection of several forms of offensive or abusive language. They enable the use of word vectors, e.g., Word2Vec (Mikolov et al., 2013), Glove (Pennington et al., 2014), or ELMo (Peters et al., 2018) instead of bag-of-words representations. Further, DL models such as Long Short Term Memory (LSTM) networks or Convolution Neural Networks (CNN) achieved significantly better results in several NLP tasks than less complex classifiers, such as Support Vector Machines, Logistic Regression or Decision Tree Models. Badjatiya et al. (2017) experimented with multiple DL architectures and text representations to detect hate speech on a dataset of 16,000 annotated English-language tweets. They demonstrated that DL approaches, in sum, outperform models based on char and word n-gram representations.

However, such models need extensive amounts of (manually labeled) training data and are often bound to the training data structure and the specific task they are trained for. As a consequence, it is problematic to apply these models to other tasks, such as detecting the outcome of the model in other languages. In our study, we applied BERT language models (Devlin et al., 2018) to detect different forms of offensive language in German tweets. BERT is not developed to solve a specific problem but constitutes a general language model. That means, BERT does not learn, e.g., what words occur in offensive tweets, but learns how words of a language (e.g. English) are generally organized and combined (Devlin et al., 2018). BERT uses an approach called "masked language model" (MLM), that allows bidirectional learning, meaning learning context both to the right and to the left of words, which previous models were not designed to do. English-language BERT models have been used in other shared tasks, such as SemEval-2019 Task 6: "Identifying and Categorizing Offensive Language in Social Media" (Zampieri et al., 2019). However, to the best of our knowledge there are no publications about German-language BERT models.

## 3 Dataset and Tasks

We trained our models on a dataset of German-language tweets provided in context of the GermEval Shared Task 2 (2019) on the identification of offensive language. The tasks consists of three classification subtasks: subtask I is a binary classification whether a tweet contains offensive language or not (coarse-grained); subtask II requires distinguishing between three subcategories of offensive language (fine-grained); and the goal of subtask III is to decide whether offensive tweets are implicit or explicit offensive. Figure 1 shows example tweets from the training data for each category.

@RoemeltA Du bist jetzt geblockt, denn rassistische Kackscheisze höre ich mir nicht an, ich lese sie nicht und noch viel weniger diskutiere ich darüber. Punkt. *OFFENSIVE*

@MiKeyyy328 schon ok ich verstehe das *OTHER*

(a) Training samples for coarse-grained classification.

@Sternenrot @_schwarzeKatze aber das ist halt einfach kein topf wtf *PROFANITY*

@Dr_Dicht Selber SCHULD, wenn Sie hässliche NAPFSÜLZE auch damit aufhören! *INSULT*

@Hallaschka_HH Antisemitismus gehört zur DNA von Luthers Kirche. *ABUSE*

(b) Training samples for fine-grained classification.

@krippmarie Ich kenne noch einige Namen unter den SPDler die ebenfalls zu Grabe getragen müssten-sollten-werden.... *IMPLICIT*

@diMGiulia1 Araber haben schon ekelhafte Fressen....! *EXPLICIT*

(c) Training samples for implicit/explicit classification.

Figure 1: Example tweets and their class labels.

### 3.1 Coarse-Grained Binary Classification

Subtask I is to decide whether a tweet includes some form of offensive language or not. For this task, 1,282 tweets labeled as *OFFENSIVE* and 2,698 tweets labeled as *OTHER* were used. Figure 1a shows examples of both categories.[1]

### 3.2 Fine-Grained Multi-Class Classification

The goal of subtask II is to detect the subcategories of offensive language, namely *PROFANITY*, *INSULT*, and *ABUSE*. *PROFANITY* is simply the usage of profane words in non-insulting contexts. *INSULT*, unlike profanity, requires an intention to offend an individual or a group. A tweet is labeled as *ABUSE* if it not only insults a person but also includes the stronger form of abusive language. In the dataset for subtask II, 152 tweets are labeled as *PROFANITY*, 624 are labeled as *INSULT* and 506 as *ABUSE*. Figure 1b shows examples of all three categories.

[1] Disclaimer: The examples may be considered profane, vulgar, or offensive. They do not reflect the views of the authors and exclusively serve to explain linguistic patterns.

### 3.3 Implicit/Explicit Classification

Subtask III aims at classifying tweets as either implicit or explicit offensive. 257 tweets are labeled as implicit and $1,664$ tweets are labeled as explicit offensive language. Figure 1c shows examples of both categories.

## 4 Fine-Tuning BERT for German Tweets

In this section, we describe our German-language BERT model and we present our simple, yet effective, ensembling strategy. Further, we detail our submitted runs.

### 4.1 BERT

The core of our approach is a case-sensitive German-language BERT model. It is pre-trained on 12 GB of raw text from the German-language Wikipedia dump, OpenLegalData dump, and news articles.[2] The model is of the same size as the English-language "BERT-Base" model (12-layers, 768-hidden, 12-heads) and comprises 110 million parameters. We use the framework FARM[3] and make our implementation available online.[4]

Tweets are padded/clipped to a maximum length of 150 tokens each. The average length in the training dataset is 41 words and less than 0.2 percent of the tweets are clipped. For fine-tuning BERT, we use a batch size of 32. Smaller training batches would most likely not contain samples from all classes. We use the Adam optimizer with an initial learning rate of $2e$-5 and warmup the learning rate on 10 percent of the training data — compared to 1 percent of the data in the original BERT paper (Devlin et al., 2018). Other parameters, such as a 10 percent embedding dropout rate, are the same as in the original paper. A weighted cross-entropy loss takes into account the unbalanced class distribution in the training data. For example, regarding the fine-grained classification subtask, the class weights are 1.96 (*ABUSE*), 6.57 (*PROFANITY*), 1.56 (*INSULT*), and 0.37 (*OTHER*). The training runs for one to four epochs, depending on the exact submission described in Section 4.3.

We optionally apply two preprocessing methods. First, we replace all user mentions, such as *@Pe_ter*, with the token *Name*. This normalization helps to reduce the variety of different user names without losing information about the entity type. Second, for the implicit/explicit classification task, both training and test set provide the true fine-grained class labels for each tweet. We append these class labels as additional text tokens at the end of each tweet to incorporate this information into our model.

BERT uses tokenized parts of words instead of tokenized words. We give two examples of this tokenization.

> text: @RobertHabeck Ihr verunglücktes Videostatement hat doch rein gar nichts mit Twitter zu tun. Sie hätten dies ja auch in irgendeine Kamera eines TV-Teams hineinsprechen können.
>
> tokens: ['Name', 'Ihr', 'ver', '##unglück', '##tes', 'Videos', '##tat', '##ement', 'hat', 'doch', 'rein', 'gar', 'nichts', 'mit', 'Twitter', 'zu', 'tun', '.', 'Sie', 'hätten', 'dies', 'ja', 'auch', 'in', 'irgend', '##eine', 'Kamera', 'eines', 'TV', '-', 'Teams', 'hinein', '##sprechen', 'können', '.']

Note that the tokenization correctly separates *hineinsprechen* into *hinein* and *sprechen, irgendeine* into *irgend* and *eine,* and *verunglücktes* into *ver, unglück* and *tes.*

> text: @Dr_Dicht Selber SCHULD, wenn Sie hässliche NAPFSÜLZE auch damit aufhören!
>
> tokens: ['Name', 'Sel', '##ber', 'SC', '##H', '##U', '##L', '##D', ',', 'wenn', 'Sie', 'hä', '##ss', '##liche', 'NA', '##P', '##FS', '##Ü', '##L', '##Z', '##E', 'auch', 'damit', 'auf', '##hören', '[UNK]']

Note that the exclamation mark at the end of the tweet is treated as an unknown symbol because the pre-trained language model discards all punctuation marks as a preprocessing step. As a consequence, our classifier does not distinguish question marks and exclamation marks and treats both as unknown symbols. Further, the tokenization does not correctly deal with words written with all capitals, such as *SCHULD*. In general, we find that uppercase letters followed by another uppercase letter are interpreted as a single token. Exceptions are abbreviations that the tokenizer learned and that are

[2] `https://deepset.ai/german-bert`

[3] `https://github.com/deepset-ai/FARM`

[4] `https://hpi.de/naumann/projects/repeatability/text-mining.html`

written with all capitals, such as *SC* for *Sportclub*. We assume that this abbreviation is learned because it frequently occurs in the pre-training data for the German BERT model, such as news articles.

### 4.2 Ensembling Strategy

Fine-tuning BERT is about tailoring the language model to a particular downstream task, such as sequence classification or question answering. An additional output layer, called prediction head, is appended to the model and trained on labeled data. The predictions of our BERT model vary when using different random weight initializations for the prediction head. Therefore, we create an ensemble of the predictions of multiple models. To this end, we use soft majority voting:

$$\hat{y} = \underset{j}{\operatorname{argmax}} \sum_{i=1}^{n} p_{i,j}$$

where $p_{i,j}$ is the probability for class label $j$ predicted by the $i$-th classifier (out of $n$ classifiers). We ensemble five runs for the binary classification tasks (coarse-grained and implicit/explicit) and ten runs for the multi-class classification task (fine-grained).

### 4.3 Submitted Runs

For each of the three tasks, we submitted three runs as described here. For the coarse-grained classification:

- `hpiDEDIS_coarse_1` one training epoch
- `hpiDEDIS_coarse_2` two training epochs
- `hpiDEDIS_coarse_3` four training epochs

For the fine-grained classification:

- `hpiDEDIS_fine_1` one training epoch
- `hpiDEDIS_fine_2` two training epochs
- `hpiDEDIS_fine_3` four training epochs

For the implicit/explicit classification:

- `hpiDEDIS_implicit_1`
  two training epochs, normalized user names
- `hpiDEDIS_implicit_2`
  four training epochs
- `hpiDEDIS_implicit_3`
  four training epochs, normalized user names

Table 1: Macro-average F1 scores on test data.

| | Ensemble | | | Single Model | | |
|---|---|---|---|---|---|---|
| Run | Coarse | Fine | Imp. | Coarse | Fine | Imp. |
| 1 | 75.3 | 42.0 | 70.8 | 74.4 | 41.0 | 70.5 |
| 2 | **76.4** | 47.1 | **73.1** | 75.5 | 45.7 | 72.1 |
| 3 | 76.1 | **51.2** | 73.1 | 75.4 | 49.1 | 72.4 |
| Base | - | - | - | 50.0 | 34.3 | 62.0 |

## 5 Results

We present the results for the identification of offensive language using BERT German-language models. We further describe our baseline approach to compare the results for each of the subtasks. Table 1 lists the macro-average F1 scores of the BERT models and the baseline approach on the test dataset for each of the three tasks. It further compares the results of the ensemble models to the single models. It can be seen that the ensemble models always outperform the corresponding single models. For example, for the fine-grained classification task, the best single model achieves a score of 49.1 compared to a score of 51.2 with the best ensemble model (4 percent improvement). All BERT based models clearly outperform the baseline models by up to 26 percentage points.

### 5.1 Baseline Approach

As a baseline approach, we applied a Logistic Regression model on a tf-idf weighted bag-of-words (BoW) feature representation of unigrams, bigrams and trigrams (Risch et al., 2018). For preprocessing, the tweets have been tokenized applying the NLTK TweetTokenizer [5], which we also used to remove Twitter handles and to normalize word tokens such as *duuuummmm* and *duuuuuuummmmmm* to a common word token *duuumm*. We further removed stopwords, using the Stopwords ISO [6] word list for German. For subtask III (implicit/explicit offensive language classification) we added the information if a tweet was abusive, profane, or insulting as a feature, which shows whether one of these forms is more likely to be expressed implicitly or explicitly.

### 5.2 Coarse-Grained

The coarse-grained classification task is the simplest of the three subtasks and our ensemble BERT

[5] `https://www.nltk.org/api/nltk.tokenize.html`

[6] `https://github.com/stopwords-iso`

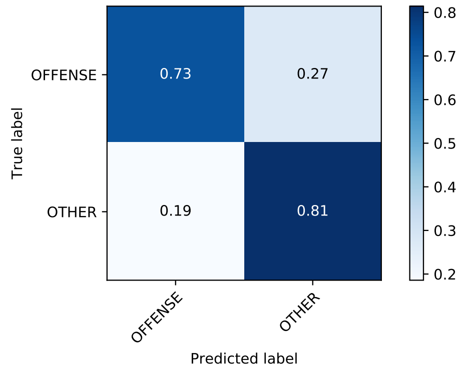
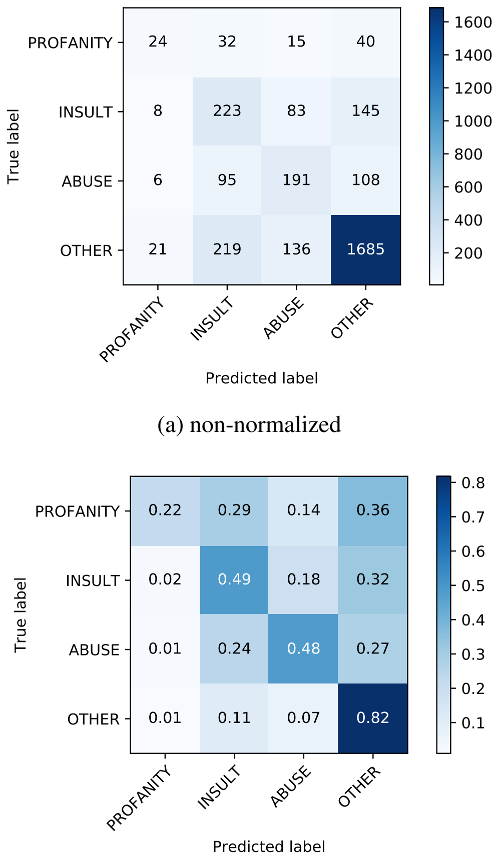


Figure 2: Normalized confusion matrix for the coarse-grained classification subtask.

model achieves a macro-average F1 score of 76.40 percent after two training epochs. Training for one epoch (75.26 percent) or four epochs (76.06 percent) yields similar performances. Figure 2 shows the normalized confusion matrix for the task. The row-based normalization discards the influence of the imbalanced class distribution so that all classes are considered to be equally important. It shows that the model is more reliable when identifying the *OTHER* class than the *OFFENSE* class. 81 percent of the non-offensive and 73 percent of the offensive tweets have been retrieved by the model. The baseline model, on the other hand, retrieved 60 percent of the offensive and 46 percent of the non-offensive tweets.

### 5.3 Fine-Grained

Figure 3 shows two confusion matrices for the fine-grained classification task. The upper subfigure uses the absolute number of samples, while the lower subfigure uses normalized numbers. The confusion matrix reveals that *OTHER* is identified most reliably by far. *INSULT* and *ABUSE* are equally well identified (recall 0.49 and 0.48). However, in percentage terms, *INSULT* is more frequently mistakenly confused with *OTHER* than with *ABUSE*. *PROFANITY* is most challenging to identify and is most frequently confused with *OTHER* and least frequently confused with *ABUSE*. This confusion matches the similarity of the classes. The baseline model struggles with the distinction of the four subcategories of offensive language and receives only 16 percent recall for PROFANITY, which was highly underrepresented in the training data. Recall for INSULT is 0.28, for ABUSE 0.26 and for OTHER 0.67.



(a) non-normalized

(b) normalized

Figure 3: Confusion matrices for the fine-grained classification subtask.

### 5.4 Implicit/Explicit

Figure 4 shows the normalized confusion matrix for subtask III on implicit and explicit offensive language classification. Implicit offensive language is way harder to identify than explicit offensive language. Most of the explicit offensive tweets are identified by the model (recall 0.92), but only about half of the implicit offensive tweets. The model makes most of the mistakes by misclassifying implicit tweets as explicit (recall 0.54). The baseline BoW-classifier for this task makes similar mistakes but performs worse on the detection of implicit offensive language. Its recall for explicit is 0.91 and 0.31 for implicit offensive language. Again, the implicit offensive language category was highly underrepresented, which probably made the model choose the explicit class when in doubt.

## 6 Conclusions and Future Work

We studied the problem of offensive language identification in the context of the GermEval task 2019.

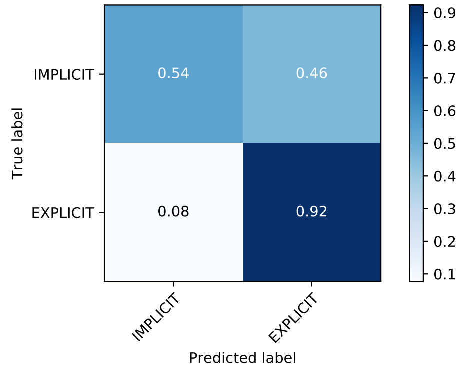


Figure 4: Normalized confusion matrix for the implicit/explicit classification subtask.

Our approach builds on a BERT model pre-trained on a large German-language corpus. To this end, we fine-tuned the model on the labeled task-specific training datasets and refrained from any feature engineering or sophisticated pre-processing. The evaluation results on the test data match the results on our validation data and the achieved macro-average F1 score beats the baseline by up to 26 percentage points. We showed that language models, such as BERT, can be successfully fine-tuned for offensive language detection for the German language.

One direction for future work on German BERT models is to find a better way for tokenization of words written in capitals. While capitalization certainly needs to be dealt with in German language models, our current model fails in recognizing words written in capitals. Another direction is the ensembling of multiple BERT models. We find that the predictions of models with differently initialized weights but trained on the same data varies. While ensembling these different models increases overall classification performance, it is unclear how this method can be leveraged best and where its limits are.

## References

Pinkesh Badjatiya, Shashank Gupta, Manish Gupta, and Vasudeva Varma. 2017. Deep learning for hate speech detection in tweets. In *Proceedings of the International Conference on World Wide Web (WWW)*, pages 759–760. International World Wide Web Conferences Steering Committee.

Kevin Coe, Kate Kenski, and Stephen A Rains. 2014. Online and uncivil? patterns and determinants of incivility in newspaper website comments. *Journal of Communication*, 64(4):658–679.

Thomas Davidson, Dana Warmsley, Michael Macy, and Ingmar Weber. 2017. Automated hate speech detection and the problem of offensive language. In *Proceedings of the AAAI Conference on Web and Social Media*.

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*, pages 1–16.

Mark Hsueh, Kumar Yogeeswaran, and Sanna Malinen. 2015. leave your comment below: Can biased online comments influence our own prejudicial attitudes and behaviors? *Human Communication Research*, 41(4):557–576.

Irene Kwok and Yuzhou Wang. 2013. Locate the hate: Detecting tweets against blacks. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems (NIPS)*, pages 3111–3119.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543.

Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, pages 2227–2237. ACL.

Julian Risch, Eva Krebs, Alexander Lser, Alexander Riese, and Ralf Krestel. 2018. Fine-grained classification of offensive language. In *Proceedings of GermEval (co-located with KONVENS)*, pages 38–44, September.

Leonie Rösner, Stephan Winter, and Nicole C Krämer. 2016. Dangerous minds? effects of uncivil online comments on aggressive cognitions, emotions, and behavior. *Computers in Human Behavior*, 58:461–470.

Nina Springer, Ines Engelmann, and Christian Pfaffinger. 2015. User comments: Motives and inhibitors to write and read. *Information, Communication & Society*, 18(7):798–815.

Marcos Zampieri, Shervin Malmasi, Preslav Nakov, Sara Rosenthal, Noura Farra, and Ritesh Kumar. 2019. SemEval-2019 task 6: Identifying and categorizing offensive language in social media (OffensEval). In *Proceedings of the International Workshop on Semantic Evaluation*, pages 75–86. ACL.