# Fine-Grained Classification of Offensive Language

Julian Risch[1], Eva Krebs[2], Alexander Löser[2], Alexander Riese[2] and Ralf Krestel[1]

Hasso Plattner Institute, University of Potsdam
[1]firstname.lastname@hpi.de
[2]firstname.lastname@student.hpi.de

## Abstract

Social media platforms receive massive amounts of user-generated content that may include offensive text messages. In the context of the GermEval task 2018, we propose an approach for fine-grained classification of offensive language. Our approach comprises a Naive Bayes classifier, a neural network, and a rule-based approach that categorize tweets. In addition, we combine the approaches in an ensemble to overcome weaknesses of the single models. We cross-validate our approaches with regard to macro-average $F_1$-score on the provided training dataset.

## 1 Toxic Comment Classification

With the ever growing popularity of the Internet, social networks nowadays have large user bases. The users of those social networks produce huge amounts of text data in form of posts. As of 2017, even if we only consider the website Twitter, there are 500 million Twitter posts (tweets) per day[1]. While the majority of those tweets uses appropriate language, there are also tweets that contain offensive language.

There are different kinds and severity levels of offensiveness. If a user describes the weather with profane words, the resulting tweet would be considered offensive. However, compared to tweets containing a direct insult or identity hate, which may even be criminal offenses, the previous example is a rather harmless offense.

Regardless of their severity, those offensive posts need to be found and moderated. Due to the high number of posts, it is not feasible to manually check each post for offensiveness. Therefore, we propose to automatically classify offensive language in tweets. In this paper, we describe a machine-learning-based approach, using ensembles of different classifiers to detect and classify different severity levels of offensive language.

## 2 Related Work

An important issue in the field of online comment classification is the availability of labeled data. Thanks to Kaggle's recent Toxic Comment Classification Challenge[2] there is a publicly available dataset of more than 150,000 comments. In this challenge participants classified Wikipedia talk-page comments at different levels of toxicity but also distinguished between obscene language, insults, threats, and identity hate. Similarly, the First Shared Task on Aggression Identification (Kumar et al., 2018) dealt with the classification of the aggression level of user posts at Twitter and Facebook. It was part of the First Workshop on Trolling, Aggression and Cyberbullying at the 27th International Conference of Computational Linguistics (COLING 2018). The task considered the three classes "overtly aggressive", "covertly aggressive", and "non-aggressive". In general, we perceive a trend towards finer-grained classification of toxic comments. Thereby the challenge shifts from detecting toxic comments to giving more specific reasons why a particular comment is considered toxic (on the basis of its subclass).

Previous research agrees that word n-grams are well-performing features for the detection of hate speech detection and abusive language (Nobata et al., 2016; Badjatiya et al., 2017; Warner and Hirschberg, 2012; Davidson et al., 2017; Schmidt and Wiegand, 2017). However, ensembles, which combine different, complementing approaches outperform single approaches and achieve especially robust results (Risch and Krestel, 2018a). Word n-grams, character n-grams, and — given a

[1]https://www.omnicoreagency.com/twitter-statistics/

[2]https://www.kaggle.com/c/jigsaw-toxic-comment-classification-challenge

large amount of training data — deep learning approaches perform well in combination.

The task of toxic comment classification is not only of theoretical significance but also has practical applications, for example at the moderation of user-generated content. It has become an industry-wide, costly challenge for online news providers to moderate their discussion platforms. To this end, different approaches have been proposed, which deal with predicting the moderation effort (Ambroselli et al., 2018) or semi-automated classification (Risch and Krestel, 2018b).

## 3 GermEval Task 2018

We consider the GermEval task 2018[3], which is to classify the offensiveness of German-language tweets. The provided training dataset consists of 5009 categorized tweets and the provided test dataset consists of 3532 uncategorized tweets. There are two tasks: (1) a coarse-grained binary classification with the categories `OFFENSIVE` and `OTHER` and (2) a fine-grained classification with the four categories `PROFANITY`, `INSULT`, `ABUSE` and `OTHER`. Both tasks are multi-class classification tasks (as opposed to multi-label classification), because the classes are mutually exclusive. In this paper, we focus on the more challenging, fine-grained classification task.

While the training data contains examples from all categories, the categories are not uniformly distributed: The majority of tweets (66.3%) is labeled `OTHER`, while `ABUSE` (20.4%) and `INSULT` (11.9%) also occur relatively often. The category `PROFANITY` is underrepresented and constitutes only 71 of the 5009 tweets (1.4%).

The category `PROFANITY`, consists of all tweets that include profane words that are not directed towards a person or group, see Figure 1a. The category `INSULT` includes tweets with negative content directed towards individuals, see Figure 1b. In contrast to the `INSULT` category, `ABUSE` encompasses negative sentiments towards social groups or their members, because of traits associated with that group, see Figure 1c. The last category, `OTHER`, contains every tweet that is not covered by the previous categories. The GermEval task is evaluated with regard to macro-average $F_1$-score, which is the unweighted mean of the $F_1$-scores of each individual category.

[3] https://projects.fzai.h-da.de/iggsa/

> @anna_IIna Kann man diesen ganzen Scheiß noch glauben..?

(a) Training sample categorized as `PROFANITY`

> @AchimSpiegel ”Sigmar Dumpfbacke Gabriel” gefällt mir richtig gut

(b) Training sample categorized as `INSULT`

> @diMGiulia1 Araber haben schon ekelhafte Fressen....!!

(c) Training sample categorized as `ABUSE`

Figure 1: Example tweets from the training dataset and their fine-grained labels.

## 4 Fine-Grained Classification of Offensive Language

We propose different approaches for the task of fine-grained classification of offensive language. These approaches are tailored to have different strengths and weaknesses. In an ensemble, we leverage that the approaches complement each other. To this end, we propose diverse approaches, such as a Naive Bayes classifier, Sentiment Polarity Lexicons, and Deep Neural Networks.

### 4.1 Naive Bayes Classifier

Our first approach uses a Naive Bayes classifier with logistic regression to categorize the tweets. Thereby the logistic regression is trained with the log-count ratios of the Naive Bayes model. Wang and Manning proved that this kind of model works very well as a baseline (Wang and Manning, 2012). Because of the underlying bag of words model, it works well with texts that contain words, more specifically bigrams, that are strong indicators for one of the categories. On the downside, it does not work well with test data that contains many unseen words.

### 4.2 Neural Network Classifier

Neural networks achieved state of the art results in different classification tasks, including Natural Language Processing centered tasks such as sentiment analysis (Zhang et al., 2018). Our network is based on an Long Short-Term Memory (LSTM) layer and a Global Maximum Pooling layer. For

the final classification, we use a Dense layer with softmax activation. The given dataset in our task is relatively small with about 5000 samples and therefore does not work well with typical deep neural networks. To solve this problem, we make use of transfer learning.

**Transfer Learning** Instead of training the network with the limited training data of the task, we pre-train the network on a related task with a larger amount of data. We use a dataset of more than 150,000 German, machine-translated from English, Wikipedia talk page comments. This dataset originates from the Kaggle Toxic Comment Classification challenge and is human-labeled with several toxicity categories. After this training phase, the weights in the neural network are adjusted to the GermEval task. Because the Kaggle challenge is similar to the GermEval task, we kept the first layers with the corresponding weights and added a shallow network of Dense layers on top of them. Afterwards, the modified network is trained on the GermEval data, whereby only the newly added Dense layers get adjusted by the backpropagation. The other weights remain unaffected with the intent to include general representations (learned on a larger dataset) in the first layers.

**Imbalanced Classes** Besides the small size of the training dataset, the distribution of the different categories is challenging in combination with the evaluation metric. In many cases, `OTHER` is wrongly predicted instead of the correct category (false positives), because this is by far the largest fraction of the training dataset and therefore often the correct result. However, the macro-average $F_1$-score takes the $F_1$-score of each category uniformly into account. This evaluation measure results in an overall bad performance if there are many false positives for the majority class.

To address this concern, we consider two approaches: class weights and generating synthetic training data with the synthetic minority oversampling technique (SMOTE) (Chawla et al., 2002). The class weights add a factor to the loss function dependent on the predicted class. In our case, this parameter was set to 'balanced' to use class weights that are inversely proportional to the class sizes and therefore increase the penalty for misclassifying minority category examples.

The SMOTE algorithm operates on the input data and generates additional samples of the minority classes in order to balance the data. This is achieved by repetitively taking samples and a number of nearest neighbors in the feature space and randomly interpolating between them. The resulting interpolation point corresponds to the newly created, additional data point for the appropriate minority class. This procedure is executed for each minority class.

### 4.3 Rule-based Classifier

The small amount of provided training data motivates to develop classifiers based on specific rules tailored to the GermEval task. For example, a tweet in the category `PROFANITY` will definitely contain a profane word, but likely not a person or group.

We collected several word lists for the rules. Some are from external sources, such as an exhaustive list of profane or insulting words, a list of German politicians and political parties, and words that are usually used in a negative context. In addition, we manually created lists with words that appeared very often in a specific context. For example, words related to the refugee crisis appeared more frequently in tweets classified as `ABUSE`.

The classifier has scores for all categories, `OTHER` being the default. The rules check for word occurrences. Each time a word is found, scores of categories related to the rule are increased. The highest score determines the predicted category.

### 4.4 Ensemble Classifier

Table 1 lists the Pearson correlation of the different classifiers' out-of-fold predictions on the training dataset. The correlation is very small, which shows that the classifiers have different strengths and weaknesses. As a consequence, they provide the opportunity to combine the individual results with an ensemble classifier, which potentially further improves predictions. We discuss two ensembling methods: logistic regression and gradient boosting trees.

**Logistic Regression and Gradient Boosting Ensembles** Due to the imbalanced class labels in the training dataset, the learning uses balancing class weights. The logistic regression approach takes only the final results of the classifiers into account. In contrast, our gradient boosting approach also considers features of the text. These features are the text length, the ratio of exclamation marks and the ratio of uppercase characters. We use a gradient boosting ensemble, in form of a light gradient

| | NB - NN | NN - RB | NB - RB |
|---|---|---|---|
| Profanity | 0.0037 | 0.0604 | -0.0052 |
| Insult | 0.0723 | 0.0235 | 0.1154 |
| Abuse | -0.0015 | 0.0809 | 0.2278 |
| Other | 0.1185 | 0.0778 | 0.2434 |

Table 1: The Pearson correlation values for each label with pairwise comparisons for Naive Bayes (NB), the neural network (NN), and the rule-based approach (RB)

boosting machine classifier (Ke et al., 2017).

### 4.5 Sentiment Polarity Lexicons

In addition to the previously described approaches, we investigate sentiment polarity lexicons, which provide a large knowledge base of word-polarity pairs. This external knowledge can potentially compensate for the relatively small amount of provided training data. Given a tweet, we infer the sentiment of each contained verb. For the classification, we consider the presence or absence of verbs with negative polarity. Further, we consider whether the negative verb refers to an entity, such as a particular person or group. Thereby, we aim to distinguish insult and abuse from profanity. We incorporate sentiment scores obtained from a variety of external sources, such as "German Polarity Cues" (Waltinger, 2010), "German Sentiment Lexicon" (Clematide and Klenner, 2010), and "SentiWS"(Remus et al., 2010). Further, we extract character n-grams and word unigrams as features for profane language based on a list of swear words.

## 5 Evaluation

As of writing this paper, the test dataset of the GermEval task is published, but not its ground truth labels. To this end, we analyze only the predicted class distribution on the test dataset. We evaluate our approaches on the provided training dataset with cross-validation.

### 5.1 Evaluation Measures

The GermEval task defines the macro-average $F_1$-score as its evaluation measure. With the measure given, we still need a set of labeled test data to evaluate our classifiers. As of writing this paper, the test dataset of the GermEval task is published, but not its labels. As a result, we can use only the training dataset as evaluation data. Since the training dataset is rather small with only 5009 labels, we decided against splitting it up in a disjoint training and test set for the evaluation. Instead, we use 5-fold cross-validation and analyze out-of-fold predictions. To this end, we split our training set into five equally sized folds. Then we choose one fold as the test set that we want to predict, and train on all other folds. We repeat this step until each fold was the test set, and thus predicted, once. This way we can predict labels for the whole test set, without ever seeing the tweets we make predictions for in the training set.

### 5.2 Discussion of the Results

Table 2 lists the evaluation results for our individual classifiers. The Naive Bayes classifier identifies most of the tweets that should be labeled `OTHER`, nearly none that are `PROFANITY` and a small amount with a relatively high precision that should be in category `INSULT` or `OTHER`. The recall of category `PROFANITY` might be especially low because this category is represented the least in the training dataset and the classifier only learns on words found in the training dataset. The opposite may be true for `OTHER`, which is the most often occurring category. In total the Naive Bayes classifier achieved an $F_1$-score of 0.366.

In comparison to the Naive Bayes classifier, the neural network detects considerably less `OTHER`, but it detects a certain amount of `PROFANITY`. The recall values for `INSULT` and `ABUSE` are also higher, but similar to `PROFANITY` they have a relatively low precision. The neural network achieved a total $F_1$-score of 0.261. This evaluation already considers our approaches against class imbalance. Both approaches, SMOTE and class weights, increased the $F_1$-score from about 0.22 to about 0.26, while the SMOTE approach performs slightly better than the class weights.

The rule-based classifier finds slightly less `OTHER` than the Naive Bayes classifier, but has a higher recall and lower precision on the other three categories. Since the rules work with very specific word lists, the classifier may be able to detect more tweets that fit the rules, but cannot differentiate them from non-offensive texts that also contain those words. The rule-based approach is the best individual classifier with an $F_1$-score of 0.390.

Our ensemble classifiers performed better than the individual classifiers: the gradient boosting ap-

| | Naive Bayes | | | Neural Network | | | Rule-based | | |
|---|---|---|---|---|---|---|---|---|---|
| | precision | recall | $F_1$ | precision | recall | $F_1$ | precision | recall | $F_1$ |
| Profanity | 0.20 | 0.01 | 0.03 | 0.02 | 0.25 | 0.04 | 0.15 | 0.28 | 0.20 |
| Insult | 0.49 | 0.13 | 0.20 | 0.17 | 0.32 | 0.22 | 0.23 | 0.21 | 0.22 |
| Abuse | 0.70 | 0.29 | 0.41 | 0.22 | 0.32 | 0.26 | 0.46 | 0.32 | 0.37 |
| Other | 0.73 | 0.97 | 0.83 | 0.78 | 0.39 | 0.52 | 0.73 | 0.81 | 0.77 |

Table 2: The $F_1$-scores for each category predicted by the Naive Bayes classifier, the neural network, and the rule-based classifier

| | Gradient Boosting Ensemble | | | Logistic Regression Ensemble | | | Sentiment Lexicons | | |
|---|---|---|---|---|---|---|---|---|---|
| | precision | recall | $F_1$ | precision | recall | $F_1$ | precision | recall | $F_1$ |
| Profanity | 0.12 | 0.51 | 0.19 | 0.17 | 0.44 | 0.25 | 1.00 | 0.03 | 0.05 |
| Insult | 0.30 | 0.43 | 0.36 | 0.43 | 0.30 | 0.35 | 0.44 | 0.29 | 0.35 |
| Abuse | 0.47 | 0.51 | 0.49 | 0.57 | 0.43 | 0.49 | 0.56 | 0.39 | 0.46 |
| Other | 0.85 | 0.70 | 0.77 | 0.80 | 0.87 | 0.83 | 0.77 | 0.90 | 0.83 |

Table 3: The $F_1$-scores for each category predicted by the gradient boosting ensemble, the logistic regression ensemble classifier, and the sentiment lexicon approach for comparison

proach reached a score of 0.450 and the logistic regression ensemble achieved a score of 0.480. Notice that no individual classifier exceeds a macro-average $F_1$-score of 0.4. The detailed results can be seen in Table3. The gradient boosting classifier has higher recall values for the three offensive categories, but a lower precision. In contrast, the logistic regression ensemble classifier has lower recall values, except for `OTHER`, but a higher precision and total score.

In context of the GermEval task 2018, the logistic regression ensemble classifier provides the best result, as it has the highest total $F_1$-score. However, if the classifiers were to be used for a real-world application (e.g. helping Twitter moderators to find tweets that they should assess), the gradient boosting approach may be better suited. The gradient boosting approach has the highest combined recall values for the three offensive labels of all our classifiers, which means that more offensive tweets would be found. In a second step, the false positives could be removed by another algorithm or a human worker.

While we cannot provide an $F_1$-score for the test set, we still use the ensemble classifiers to predict its labels. We also use out-of-fold predictions, but instead of predicting for the remaining fold, we predict the entire test set. The result of this procedure are five complete prediction files, which are later combined into a final prediction by calculating the average.

The gradient boosting ensemble predicts more tweets to be in the three offensive categories. In contrast, the logistic regression approach classifies more tweets as `OTHER`. We assume that the samples' ground truth categories follow the same frequency distribution in the training set and the test set. The general category distribution of both classifiers' predictions is similar to the distribution of the categories in the training data. The `OTHER` category occurs the most often and `PROFANITY` the least often, which is shown in Figure 2. However, the distribution of the training set and the predictions for test set do not match exactly. This discrepancy is an opportunity for more optimization, which goes beyond this paper.

### 5.3 Test Dataset Submission

We submitted prediction files for the two tasks of fine-grained and coarse-grained classification. The logistic regression ensemble, the sentiment polarity lexicons, and a combination of both approaches comprise our final submission. The combination is the mean of the predicted probabilities of both approaches. The files correspond to our previously described approaches as follows:

- `hpiTM_fine_1.txt`: logistic regression ensemble

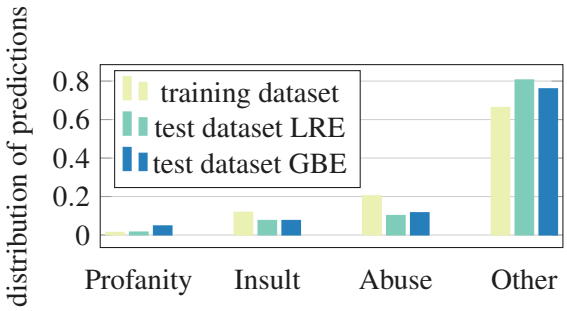


Figure 2: Category distribution predicted by the logistic regression ensemble (LRE) and gradient boosting ensemble (GBE) for the 3532 tweets in the test dataset compared with the training distribution of 5009 tweets

- `hpiTM_fine_2.txt`: sentiment polarity lexicons
- `hpiTM_fine_3.txt`: sentiment polarity lexicons and logistic regression ensemble combined
- `hpiTM_coarse_1.txt`: logistic regression ensemble

## 6 Conclusion

In this paper we considered the problem of classifying German tweets into four different categories of offensive language in context of the GermEval task 2018. This task uses the macro-average $F_1$-score as evaluation measure. In order to maximize this score, we proposed different classifiers, such as a Naive Bayes classifier, a neural network, and a rule-based approach. The results of these classifiers were combined in two different ensemble methods to achieve a higher score. This ensemble achieves a macro-average $F_1$-score of 0.48 at cross-validation on the provided training dataset. We provide our source code online[4].

An interesting path for future work is to provide fine-grained classification labels to content moderation teams at online platforms. The fine-grained labels can provide an explanation for why a particular user comment is considered toxic and may be deleted by the moderation team. To this end, even finer-grained labels that describe the target group of an insult, such as a particular religion, ethnic minority or nationality are needed. Based on such labels, also an analysis of offensive language could go into more detail and shine a light on reasons for and intentions of toxic comments.

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

## Acknowledgments

We thank Samuele Garda for his help with this project and for his valuable feedback.

## References

Carl Ambroselli, Julian Risch, Ralf Krestel, and Andreas Loos. 2018. Prediction for the newsroom: Which articles will get the most comments? In *Proceedings of the 16th Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, pages 193–199. ACL, June 1–6.

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.

Nitesh V Chawla, Kevin W Bowyer, Lawrence O Hall, and W Philip Kegelmeyer. 2002. Smote: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, 16:321–357.

Simon Clematide and Manfred Klenner. 2010. Evaluation and extension of a polarity lexicon for german. In *Proceedings of the First Workshop on Computational Approaches to Subjectivity and Sentiment Analysis*, pages 7–13.

Thomas Davidson, Dana Warmsley, Michael Macy, and Ingmar Weber. 2017. Automated hate speech detection and the problem of offensive language. In *Proceedings of the International Conference on Web and Social Media (ICWSM)*, pages 512–515.

Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. 2017. Lightgbm: A highly efficient gradient boosting decision tree. In *Advances in Neural Information Processing Systems*, pages 3146–3154.

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

Chikashi Nobata, Joel Tetreault, Achint Thomas, Yashar Mehdad, and Yi Chang. 2016. Abusive Language Detection in Online User Content. In *Proceedings of the International Conference on World Wide Web (WWW)*, pages 145–153. International World Wide Web Conferences Steering Committee.

Robert Remus, Uwe Quasthoff, and Gerhard Heyer. 2010. Sentiws - a publicly available german-language resource for sentiment analysis. In *Proceedings of the Conference on International Language Resources and Evaluation (LREC)*. European Languages Resources Association.

Julian Risch and Ralf Krestel. 2018a. Aggression identification using deep learning and data augmentation. In *Proceedings of the First Workshop on Trolling, Aggression and Cyberbullying (co-located with COLING)*, pages 150–158, August.

Julian Risch and Ralf Krestel. 2018b. Delete or not delete? semi-automatic comment moderation for the newsroom. In *Proceedings of the First Workshop on Trolling, Aggression and Cyberbullying (co-located with COLING)*, pages 166–176, August.

Anna Schmidt and Michael Wiegand. 2017. A Survey on Hate Speech Detection Using Natural Language Processing. In *Proceedings of the International Workshop on Natural Language Processing for Social Media (SocialNLP)*, pages 1–10. ACL.

Ulli Waltinger. 2010. Germanpolarityclues: A lexical resource for german sentiment analysis. In *Proceedings of the Conference on International Language Resources and Evaluation (LREC)*. European Language Resources Association.

Sida Wang and Christopher D Manning. 2012. Baselines and bigrams: Simple, good sentiment and topic classification. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Short Papers-Volume 2*, pages 90–94. Association for Computational Linguistics.

William Warner and Julia Hirschberg. 2012. Detecting hate speech on the world wide web. In *Proceedings of the Workshop on Language in Social Media (LSM)*, pages 19–26. ACL.

Lei Zhang, Shuai Wang, and Bing Liu. 2018. Deep learning for sentiment analysis: A survey. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, page e1253.