Proceedings of the 15th Conference on Natural Language Processing (KONVENS 2019)

FraunhoferSIT at GermEval 2019: Can Machines Distinguish Between Offensive Language and Hate Speech? Towards a Fine-Grained Classification

Inna Vogel **Roey Regev**
Fraunhofer Institute SIT
Rheinstrasse 75
64295 Darmstadt
inna.vogel@sit.fraunhofer.de

Abstract

In this paper, we describe the Fraunhofer-SIT submission for the *"GermEval 2019 – Shared Task on the Identification of Offensive Language"*. We participated in two subtasks: task 1 is a binary classification of German tweets on the identification of offensive language. Task 2 is a fine-grained classification to distinguish between three subcategories of offensive language. Our best model is an SVM classifier based on tf-idf character n-gram features. Our submitted runs in the shared task are: *FraunhoferSIT_coarse_[1-3].txt* for task 1 and *FraunhoferSIT_fine_[1-3].txt* for task 2. Our final system reaches 0.70 macro-average F_1-score for the binary classification and 0.46 F_1-score for the fine-grained classification. The achieved results show that the problem of automatically distinguishing between offensive language and "Hate Speech" is far from being solved.

1 Introduction

In the pseudonymous environment of social media and due to the massive rise of user-generated content on the internet, hate speech and offensive language are easily produced and spread. As the amount of harmful comments and posts is continuously growing, it is not feasible to manually check each text message for suspicious content. Additionally, hate speech violates more than just feelings. It can be extremely harmful to society, for example by inciting mass violence. Governments and social network platforms can benefit from automatic detection and prevention of malicious posts on the net (Fortuna and Nunes, 2018).

But what constitutes hate speech and when does it differ from offensive language? A unified definition does not exist yet. The consensus is that hate speech targets disadvantaged social groups in a manner that is potentially harmful to them (Jacobs and Potter, 2001). Many studies still tend to conflate hate speech and offensive language. A key challenge for automatic hate speech detection on social media is the separation of hate speech from other instances of offensive language (Davidson et al., 2017).

As there is a pressing demand for methods to automatically identify hate speech and other suspicious posts, we participate in this year's *"GermEval Task 2019 – Shared Task on the Identification of Offensive Language"*[1]. The task is focused on detecting offensive comments in a set of German tweets in three subtasks. Task 1 is a binary classification to distinguish offensive from non-offensive tweets. Task 2 requires a more fine-grained classification of offensive tweets which are divided into three subcategories - profanity, insult and abuse (Ruppenhofer et al., 2018). The first category "PROFANITY" uses insults and swear words, but not against a person or a group. Tweets that are labeled as "INSULT" want to offend someone. We categorize "ABUSE" as hate speech as it uses "language that is used to express hatred towards a targeted group or is intended to be derogatory, to humiliate, or to insult the members of the group" and can be defined according to Davidson et al. (2017) as hate speech. Subtask 3 focuses on the classification of explicit and implicit offensive language.

We participated in task 1 and 2 of the competition. In this paper, we report on the FraunhoferSIT system to classify German tweets with respect to their offensiveness. First, we give a short overview of the work already conducted in the field of offensive language detection. In Section 3, we describe the competition tasks and the data provided by the GermEval organizers. Section 4 is dedicated to the

[1] Germeval Task 2, 2019: https://projects.fzai.h-da.de/iggsa/

machine learning methodology used. We describe the text preprocessing and the features we used to train our SVM models. In Section 5, we evaluate the performance of our approaches. Lastly, we conclude our paper in Section 6.

2 Related Work

A wide range of machine learning and deep-learning approaches have been implemented to automatically detect offensive language in text data. Schmidt and Wiegand (2017) provide in their survey a short and comprehensive overview of automatic hate speech detection with a focus on feature extraction. Using bag of words or embeddings can yield to reasonable classification performance. In their survey, they outlined that character-level-approaches perform better than token-level-approaches because they reduce the spelling variation problem often faced when working with user-generated content. Lexical resources, such as a list of slurs, can also help to improve the performance but only if they are combined with other types of features.

Nobata et al. (2016) combine lexical features such as n-grams, as well as syntactic features with distributional semantics to detect abusive language in English comments on “Yahoo! Finance and News”. Badjatiya et al. (2017) and Davidson et al. (2017) also confirm that word n-grams are well-performing features for the detection of hate speech and abusive language.

Supervised approaches like Logistic Regression (Djuric et al., 2015; Del Vigna et al., 2017) or Support Vector Machines (Del Vigna et al., 2017; Davidson et al., 2017) have shown to obtain good results in classifying abusive language. Del Vigna et al. (2017) trained their SVM with word embeddings and Davidson et al. (2017) used bigram, unigram and trigram features, each weighted by its tf-idf. LSTM (Del Vigna et al., 2017; Badjatiya et al., 2017) and Convolutional Neural Network classifiers employed on word embeddings or other pretrained representations of words and tokens are also highly effective for the task of classifying abusive language (Badjatiya et al., 2017; Ho Park and Fung, 2017).

Most of the research is focused on English datasets. The identification of toxicity in German language messages received less attention by the researchers so far. Comparable cross-lingual research is sparse.

3 GermEval Competition Task 2019

The organizers of GermEval 2019 provided training and test datasets[2] for three offensive language detection tasks. The provided datasets for training consist of 12,536 tweets without any user metadata. Task 1 is a binary classification for deciding whether a German tweet contains offensive language (the category labels and the number of tweets are “OFFENSE”: 4,177 and “OTHER”: 8,359). Task 2 is a multi-class classification and distinguishes between three subcategories of offensive language with the more fine-grained labels “PROFANITY”: 271, “INSULT”: 1,601 and “ABUSE”: 2,305. While training data contains examples of all categories, the class distribution is fairly imbalanced. The majority of tweets are neutral (67%). Abusive tweets are also relatively often (18.4%), while the class “PROFANITY’ is underrepresented (2.2%). The three classes that contain offensive language are defined as follows (Ruppenhofer et al., 2018):

- **Profanity:** abusive words are used but the tweet does not insult someone (e.g. *“ich scheiß auf deine Gedenkkultur sie geht mir am Arsch vorbei”*)
- **Insult:** profanity is directed at an individual with the intention to insult the person or group (e.g. *“SPD ihr seit wirklich das Asozialste Pack”*)
- **Abuse:** the tweet is intended to demean and attack another person or a group with cruel and derogatory language which we categorize as hate speech (e.g. *“Der verdammte Drecksack und Massenmörder Israel will mit allen Mitteln ein Krieg gegen Russland”*)
- **Other:** the tweet is neutral and does not contain any assertive or offensive words (e.g. *“Ach so vergessen , einen schönen guten Morgen”*)

Subtask 3 focuses on the classification of explicit and implicit offensive language. Explicit tweets directly express hatred towards a particular target. With implicit tweets, hatred of a target must be derived from the context. In this paper, we focus on the binary classification (task 1) as well as the more challenging fine-grained classification task 2.

[2] GermEval 2019 Data and Tasks: `https://projects.fzai.h-da.de/iggsa/projekt/`

The competition results are evaluated with regard to macro-average F_1-score, which is the unweighted mean of the F_1-scores of each individual category.

4 Methodology

In the following, the same approach is applied to both classification tasks. First, the Twitter data was preprocessed to handle idiosyncrasies such as hashtags, Emojis or irrelevant characters. Afterwards, character n-grams are extracted as features which serve as tf-idf weighted input to train a Support Vector Machine (SVM).

4.1 Preprocessing

Before extracting features and training the SVM on the tweets, we perform different pre-processing techniques according to the following procedure:

1. Removing unnecessary white spaces.
2. Lowercasing all characters.
3. Smileys and Emojis are replaced with CLDR (Common Locale Data Repository) - short character names and keywords.
4. Replacing question marks with the placeholder <*question*> and numbers with the placeholder <*number*>.
5. Deleting irrelevant symbols and characters, e.g. "+,*,/,".
6. Sequences of the same characters with a length greater than three are removed.
7. Removing words with less than three characters.
8. Removing stopwords by using the NLTK (Natural Language Toolkit) list of stopwords for the German language.
9. To tokenize the words we used the TwitterTokenizer from the NLTK library. The TwitterTokenizer is adapted for Twitter and other forms of casual speech used in social networks. It contains some regularization and normalization features (e.g. converting tweets to lowercase and vice-versa, removing username mentions and reducing the length of words in the tweet with repeated characters).
10. Hashtags (#) and the attached token were processed in a special way since they are widely used on Twitter and contain semantic content. First, the hash sign was removed. Compound words were separated in a sequence of meaningful terms (e.g. "#NoAfD" - "No AfD", "#KölnerTreff" - "Kölner Treff", "#ichwars" - "ich wars", "#OSZE-Beobachter" - "OSZE Beobachter", "#EU-Beitrittsgespräche" - "EU Beitrittsgespräche").

We also experimented with replacing all hashtags and @-Mentions with placeholders (<*HashTag*>, <*AM*>). Since this had no impact on the performance of the classifier, we discarded this preprocessing step from the final submission.

4.2 Features

After preprocessing, we used scikit-learn's[3] term frequency-inverse document frequency (tf-idf) weighting function (*tfidfVectorizer*) to convert the tokens to a matrix of tf-idf features in order to build a vector pipeline (Pedregosa et al., 2011). The following n-gram features have been tested for both classifiers:

a) word n-grams with n $\in \{1,2,3\}$

b) character n-grams with n $\in \{1,2,..,7\}$

c) Feature union of word and character n-grams

The best performance was achieved by training b). When building the vocabulary, terms that have a document frequency lower than 2 were ignored.

4.3 SVM Implementation

To train the two classifiers, we used scikit-learn's SVM with tf-idf character n-grams in the range 1 to 7 as features. Task 1 is a binary classification with the classes "OTHER" and "OFFENSE". Task 2 is a multi-class classification problem with the four classes "OTHER", "PROFANITY", "INSULT" and "ABUSE". For both models OVR ("One-vs.-Rest") was used as a decision function. To solve a multiclass classification task, OVR combines multiple binary SVM classifiers (Huang et al., 2005). Each SVM classifies samples into the corresponding class against all the other classes (Hong and Cho, 2006).

When experimenting with the training set, we split the data provided by the organizers into two

[3] http://scikit-learn.org

parts. For training, we used 70% of the data. The remaining 30% were used as test set. We experimented with hyperparameter tuning, manually and by employing scikit-learn's grid search function. The performance improved slightly by using Sigmoid Function instead of the Radial Basis Function (RBF) kernel.

For the fine-grained classification (task 2) we initially trained two models. The first model is a binary classifier (the same approach as used for task 1) distinguishing between the two classes "OTHER" and "OFFENSE" - the latter containing all three of the abusive language classes described above. The second model was trained to differentiate between the three classes of offensive language "PROFANITY", "INSULT" and "ABUSE". In a second approach to task 2, a multi-class SVM was used, which was trained on all four classes simultaneously. Our second approach achieved slightly better performance results.

Finally, each model was performed on the official GermEval 2019 test set. For every task we submitted three runs, each tested with different data sets. The first run was trained on the entire dataset provided by the GermEval organizers. The second was trained only on this year's dataset (2019), the third on text data from 2018. The best classification results in both tasks were achieved using the whole data to train the model. The classification results are provided in the following section.

5 Evaluation

In this chapter, we report on runs using our SVM models trained on the data provided by the organizers. The outputs of the SVM classifiers were submitted to the GermEval competition under the submission name **FraunhoferSIT** (see Table 1).

Submitted runs (name)	**Dataset**	**Task**
FraunhoferSIT_coarse_1.txt	2018/19	Task 1
FraunhoferSIT_coarse_2.txt	2019	Task 1
FraunhoferSIT_coarse_3.txt	2018	Task 1
FraunhoferSIT_fine_1.txt	2018/19	Task 2
FraunhoferSIT_fine_2.txt	2019	Task 2
FraunhoferSIT_fine_3.txt	2018	Task 2

Table 1: Submitted runs by FraunhoferSIT

We participated in task 1, the binary classification task distinguishing offensive from non-offensive tweets and the more challenging fine-grained classification task 2. For each task, we submitted three runs.

The GermEval task defines the macro-average F_1-score as its evaluation measure. The results of Tasks 1 and 2 have shown that there are only minor differences in performance. Our best model, an SVM classifier, using tf-idf weighted character n-gram features on the entire training data provided by the organizers achieved the best performance among the three runs submitted for the tasks. For task 1 a macro F_1-score of 0.70 could be achieved. Our best performance for task 2 was a F_1-score of 0.46. In our experiments, character n-grams outperformed token n-gram features. The best n-grams at the character level range from 1 to 7. We expect our model's performance to improve further with another set of features and more training data. As shown in our submitted runs, this would be particularly helpful for the second, fine-grained task, where our classifiers performed really poorly. The performance results for the best runs of task 1 and 2 are displayed in the following tables 2 and 3.

Task 1: SVM with tf-idf character n-grams			
Category	**Performance Measure**		
	P	R	F_1
Other	81.24	78.62	80.42
Offense	58.46	60.93	59.67
Average	**69.85**	**70.27**	**70.06**

Table 2: Official evaluation results for each category of task 1 with the metrics Precision (P), Recall (R), and F_1

Task 2: SVM with tf-idf character n-grams			
Category	**Performance Measure**		
	P	R	F_1
Other	81.37	77.58	79.43
Profanity	50	9.01	15.27
Insult	34.74	39.43	36.94
Abuse	33.52	44	38.05
Average	**49.91**	**42.51**	**45.91**

Table 3: Official evaluation results for each category of task 2 with the metrics Precision (P), Recall (R), and F_1

6 Conclusion

In this paper, we have described the system submitted to the *"Shared Task on Identification of Offensive Language GermEval 2019"* by FraunhoferSIT

(Darmstadt). We participated in two tasks. The first task is a binary classification of German tweets on the identification of offensive language with the two classes “OFFENSE” and “OTHER”. The second task was a more challenging fine-grained classification. In addition to detecting offensive tweets, the task was to distinguish between the three subcategories: “PROFANITY”, “INSULT” and “ABUSE”. We trained an SVM as part of scikit-learn’s library and used tf-idf character n-grams in the range 1 to 7 as features. Our model achieves a macro F_1-score on task 1 of 0.70 and on task 2 of 0.46. The relatively poor results, especially for task 2, show that our system cannot reliably distinguish between offensive language and hate speech and that more research needs to be done to improve the classification performance.

Acknowledgments

This work was supported by the German Federal Ministry of Education and Research (BMBF) under the project “X-SONAR” (Extremist Engagement in Social Media Networks: Identifying, Analyzing and Preventing Processes of Radicalization).

References

Pinkesh Badjatiya, Shashank Gupta, Manish Gupta, and Vasudeva Varma. 2017. Deep learning for hate speech detection in tweets. In *Proceedings of the 26th International Conference on World Wide Web Companion*, WWW ’17 Companion, pages 759–760, Republic and Canton of Geneva, Switzerland. International World Wide Web Conferences Steering Committee.

Thomas Davidson, Dana Warmsley, Michael W. Macy, and Ingmar Weber. 2017. Automated hate speech detection and the problem of offensive language. In *ICWSM*.

Fabio Del Vigna, Andrea Cimino, Felice Dell’Orletta, Marinella Petrocchi, and Maurizio Tesconi. 2017. Hate me, hate me not: Hate speech detection on facebook. 01.

Nemanja Djuric, Jing Zhou, Robin Morris, Mihajlo Grbovic, Vladan Radosavljevic, and Narayan Bhamidipati. 2015. Hate speech detection with comment embeddings. In *Proceedings of the 24th international conference on world wide web*, pages 29–30. ACM.

Paula Fortuna and Sérgio Nunes. 2018. A survey on automatic detection of hate speech in text. *ACM Comput. Surv.*, 51(4):85:1–85:30, July.

Ji Ho Park and Pascale Fung. 2017. One-step and two-step classification for abusive language detection on twitter. 06.

Jin-Hyuk Hong and Sung-Bae Cho. 2006. Multi-class cancer classification with ovr-support vector machines selected by naïve bayes classifier. In Irwin King, Jun Wang, Lai-Wan Chan, and DeLiang Wang, editors, *Neural Information Processing*, pages 155–164, Berlin, Heidelberg. Springer Berlin Heidelberg.

D.S. Huang, X.P. Zhang, and G.B. Huang. 2005. *Advances in Intelligent Computing: International Conference on Intelligent Computing, ICIC 2005, Hefei, China, August 23-26, 2005, Proceedings*. Number Teil 1 in Lecture Notes in Computer Science. Springer Berlin Heidelberg.

James B. Jacobs and Kimberly Potter. 2001. *Hate crimes: criminal law and identity politics*. Oxford University Press New York.

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*, WWW ’16, pages 145–153, Republic and Canton of Geneva, Switzerland. International World Wide Web Conferences Steering Committee.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Édouard Duchesnay. 2011. Scikit-learn: Machine learning in python. *J. Mach. Learn. Res.*, 12:2825–2830, November.

Josef Ruppenhofer, Melanie Siegel, and Michael Wiegand. 2018. Guidelines for iggsa shared task on the identification of offensive language. *March*, 12:2018.

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.