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

The HUIU Contribution for the GermEval Shared Task 2

Melanie Andresen, Melitta Gillmann, Jowita Grala, Sarah Jablotschkin,
Lea Röseler, Eleonore Schmitt[†], Lena Schnee, Katharina Straka,
Michael Vauth, Sandra Kübler[‡], Heike Zinsmeister
Universität Hamburg, [†]Universität Bamberg,[‡]Indiana University
{melanie.andresen,melitta.gillmann,sarah.jablotschkin}@uni-hamburg.de
{Jowita.Grala,Lea.Roeseler}@studium.uni-hamburg.de
{Lena.Schnee,Katharina.Straka}@studium.uni-hamburg.de
{michael.vauth,heike.zinsmeister}@uni-hamburg.de
eleonore.schmitt@uni-bamberg.de, skuebler@indiana.edu

Abstract

In this paper, we present the HUIU system (a collaboration of University of Hamburg and Indiana University) for the GermEval 2019 shared task 2, subtask 1 – the coarse-grained classification of tweets into the classes OFFENSE or OTHER. Our system uses linear SVMs with character n-grams ($5 \leq n \leq 10$), POS n-grams ($3 \leq n \leq 9$) and the tweet's length in number of tokens as features. We obtain a macro-averaged F-score of 65.32 on the test data.

1 Introduction

In this paper, we report on the HUIU team's submission to the *GermEval Task 2, 2019 - Shared Task on the Identification of Offensive Language*. Three subtasks were offered. Subtask I was a binary classification task and required discriminating offensive from non-offensive tweets. Subtask II consisted of a more fine-grained classification: Each of the offensive tweets had to be marked with one of the following labels: PROFANITY, INSULT, ABUSE. Subtask III required labeling the offensive tweets as explicitly or implicitly offensive. We participated in Subtask I, i.e., the detection of offensive language in binary classified twitter data.

Our contribution is the result of a class project conducted at the University of Hamburg. The authors participated in a 6-day compact class that provided an introduction to machine learning for linguistics and digital humanities, under the supervision of Kübler and Zinsmeister. Most of the participants had basic knowledge in programming, but no experience with machine learning. The class was structured to provide a practical introduction to machine learning. Therefore, the Shared Task offered a good opportunity to familiarize the participants with every step in the process of translating a problem into a machine learning problem, deciding on a machine learning algorithm, specifying feature sets, extracting features, and training the machine learning algorithm. In addition to this task, the group also participated in GermEval 2019 Task 1 on hierarchical classification of blurbs (Andresen et al., 2019).

Using the python library scikit-learn (Pedregosa et al., 2011), we tested different models and features for the binary classification task of identifying offensive tweets. A bag-of-words approach which employs a linear SVM classifier using character n-grams combined with additional features yielded the best results.

The rest of the paper is structured as follows: We will briefly present the best systems of last year's GermEval Shared Task as well as this year's SemEval Shared Task in the section 2. In section 3, we will describe the experimental setup, i.e., the data of our Shared Task, how we preprocessed the tweets, which features we extracted, which classifier algorithm, implementation, and evaluation we used for our experiments. The best scores that we achieved during development are presented together with the final test scores in section 4. For our overall ranking in the Shared Task we have to refer to the summary published at the Workshop in Erlangen 2019. At the time of submitting this paper we did not have this information. In section 5, we will conclude our paper with a short summary and outlook on additional features and methods that we have not taken into account.

2 Related Work

The GermEval 2019 task on the Identification of Offensive Language is the second edition of the original task from 2018 (Wiegand et al., 2018). This year's task is based on a different data set than last year's.[1] But parallel to 2018, Subtask I requires a binary classification system that discriminates between offensive and non-offensive tweets. In 2018 the team that reached the highest results (Montani and Schüller, 2018) extracted as features character *n*-grams, stemmed token *n*-grams, the TF/IDF scores of both feature classes, and word embedding vectors as features. The TF/IDF scores of the token *n*-grams proved to be their most important features, i.e., removing those from the model caused a large drop of the F1-score. Their classification system was an ensemble of supervised learning methods (Logistic Regression and Random Forests) implemented in scikit-learn (Pedregosa et al., 2011). Montani and Schüller (2018) reported worse results using deep learning models (LSTM, CNN, Convolution+GRU).

The GermEval shared tasks are the German equivalent of the SemEval-2019 Task 6 (OffensEval) Subtask A (Zampieri et al., 2019), which uses a data set consisting of English tweets. The data set is also remarkably larger than the one used in the GermEval 2018 task: It comprises more than 14,000 tweets (Zampieri et al., 2019), as compared to the approximately 5,000 tweets in the 2018 GermEval task. In contrast to the results reported by Montani and Schüller (2018), the best performing team at OffensEval Subtask A used a deep learning model – BERT (Devlin et al., 2018), based on bidirectional training of the attention model Transformer (Liu et al., 2019).

As discussed in the introduction, our goal of participating in the Shared Task was to acquire a basic understanding of machine learning in the setting of a compact introductory course. Therefore, we chose a common and easy to adapt SVM algorithm with a number of features, as described in the following section, and did not take into consideration prior more elaborated attempts at approaching this specific machine learning problem.

3 Experimental Setup

3.1 Data Set

We used the training and test data sets provided by the Shared Task, which consisted of 3,995 manually annotated tweets[2]. Each tweet was labeled either as OFFENSE or as OTHER. On the second annotation level, each of the tweets of the category OFFENSE was marked with one of the following more fine-grained labels: ABUSE, PROFANITY, INSULT. Subtask I, in which we participated, did only take the binary coarse-grained labels into account. Examples (1) to (3) show example tweets from the annotated data set provided by the Shared Task. The annotation guidelines can be found online in the repository of the Shared Task 2018[3].

	OFFENSE	OTHER	**sum**
train	1,141	2,455	3,596
dev	146	253	399
sum	1,287	2,708	3,995

Table 1: Distribution of OFFENSE and OTHER in our training and development splits.

(1) @DrDavidBerger Die wirklichen Rassisten sitzen in der GroKo und bei den Grünen
('@DrDavidBerger The real racists are sitting in the GroKo and in the Green Party')
OFFENSE ABUSE

(2) Sein Charakter war ihm wichtiger anstatt als billige Nute für Korrupte Regierungen zu arbeiten .Er hat das Leben begriffen
('His character was more important to him instead of working as cheap whore for corrupt governments .He understands life')
OFFENSE PROFANITY

(3) @de_sputnik Eine Weltherrschaft führt zum Krieg bis zum bitteren Ende
('@de_sputnik World domination leads to war to the bitter end')
OTHER OTHER

In order to optimize our system, we split the provided training set into 90% for our actual train(ing) set and 10% for our dev(elopment) set by taking every tenth instance for development. Table 1 shows the distribution of tweets and coarse-grained labels in the train and dev set respectively. For the final submission, we trained the system on the complete training set.

3.2 Extracted Features

For preprocessing, we tokenized and part-of-speech (POS) tagged the data. We used the python

[1] The data set from 2018 was also available though.

[2] As mentioned above, the data set from 2018 was also available. We decided against using this additional set since we did not know if the data were different in distribution.

[3] `https://github.com/uds-lsv/GermEval-2018-Data/blob/master/guidelines-iggsa-shared.pdf`

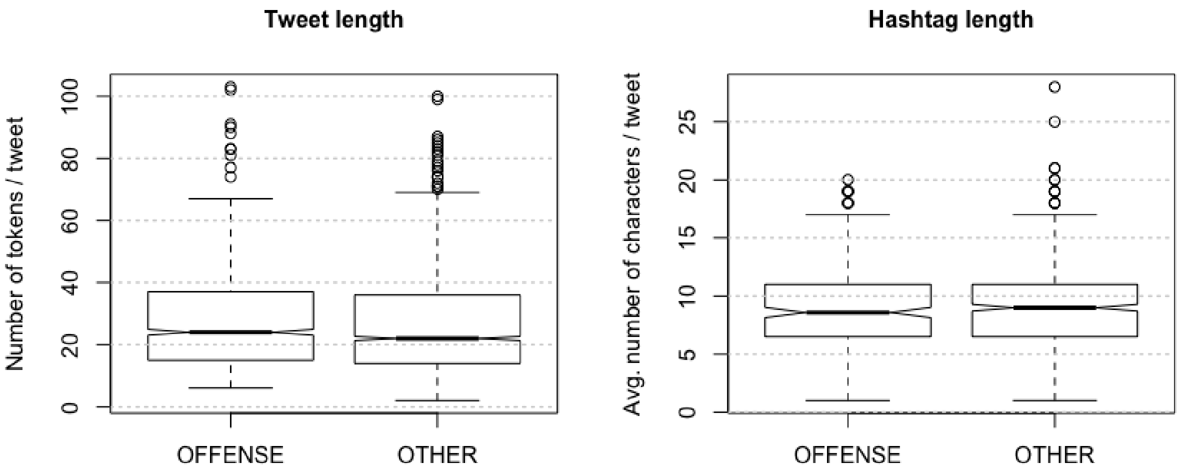

Figure 1: Boxplots of the distributions of tweet and (average) hashtag lengths per tweet in the 2019 training data (all tweets: n=3,995, tweets w/ hashtags: n=1,024; medians are marked by bold lines and notches).

implementation[4] of twokenizer, a tokenizer especially designed for twitter data (Owoputi et al., 2013). For POS tagging, we employed *TnT* (Brants, 1998), trained on the Tübingen Treebank of Written Language (Tüba-D/Z) (Telljohann et al., 2006), version 10, assigning STTS labels (Schiller et al., 1999).

As basic features, we extracted token and POS *n*-grams for bag-of-words approaches of various classifiers, see Section 3.3 for the description of the classifiers.

We found that for our final system the bag-of-words approach was most effective when using character *n*-grams of the tokens (crossing word boundaries) combined with POS *n*-grams. For the tokens, the best results were achieved with a range of 5-10 characters; for the POS tags, a range of 3-9 words led to the best results.

In order to identify additional features that help to improve our model, we extracted further textual features per tweet. First, we counted the number of tokens per tweet as well as the number of @'s and #'s. Our hypothesis was that the emotional language of offensive tweets differed from other tweets in length and in the number of addressing terms and hashtags. In addition, we tested the length of hashtags because we assumed that in emotional tweets writers tend to use hashtags consisting of longer phrases or even full sentences. Therefore, we also determined the mean length of hashtags. Since we assumed that offensive tweets might be characterized by a specific use of punctuation marks and their combinations, we counted the occurrences of the following stand-alone punctuation marks . , *! ?* as well as sequences of more than one of the same or different of these punctuation marks. We added an *n*-gram analysis of the (unicodes of) emojis extracted from the data, which we tested on character as well as on word basis.

Figures 1 and 2 illustrate the distribution of most of these features in the 2019 training data. In Figure 1, the boxplot on the left-hand side shows that offensive tweets are in fact significantly longer on average than other tweets.[5] This does not hold true for the average lengths of the strings starting with #, see the boxplot on the right-hand side.[6] Also the number of hashtags per tweet is on average slightly higher in other tweets than in offensive ones contrary to our expectations—if there are any hashtags at all, see the barplot for hashtag frequency in Figure 2.

With respect to addressing expressions, Figure 2 shows that offensive tweets tend to have in fact slightly more such elements than other tweets.[7]

[4] `https://github.com/myleott/ark-twokenize-py`

[5] Tweet length: OFFENSIVE: median=24.00, mean=26.58; OTHER: median=22.00, mean=25.67; Wilcoxon rank sum test: W=1827400, $p<0.05$.

[6] Hashtag length: OFFENSIVE: median=8.58, mean=8.88; OTHER: median=9.00, mean=8.85; Wilcoxon rank sum test: W=80980, p=0.9021.

[7] Addressing expressions: OFFENSIVE: median=1,

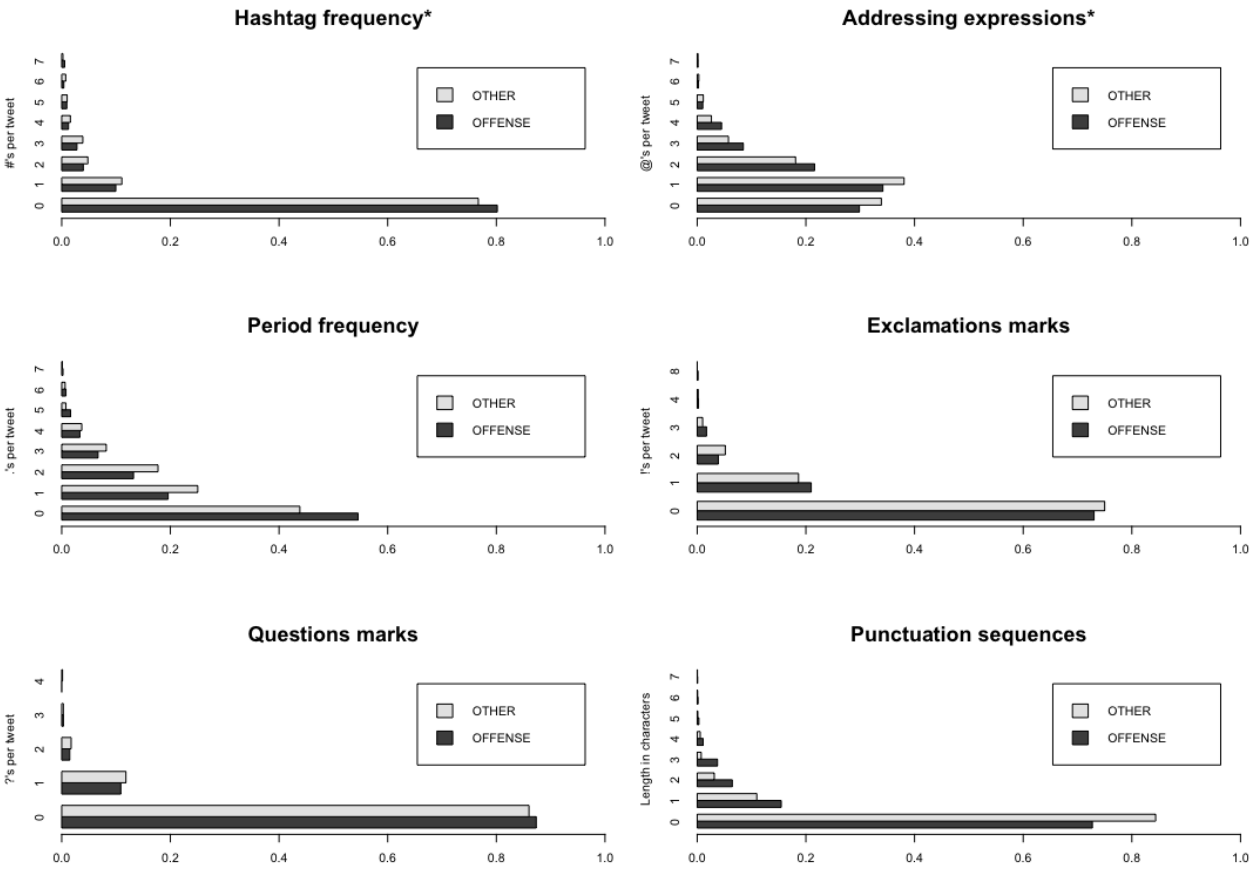

Figure 2: Additional feature distributions in the 2019 training data (n=3,995, cf. Table 1). X-axes: relative frequency of tweets. *) Numbers per tweet are cut off for better display: y<8 .

The same holds for sequences of punctuation marks (see bottom right of Figure 2). In contrast to addressing expressions, punctuation sequences are sparse in the data: About 73% of the offensive tweets and 84% of the other tweets do not contain any sequence of punctuation. The other barplots in Figure 2 show that data sparseness also holds for the distributions of exclamation marks and question marks, slightly less for periods which are on average more frequent in other tweets than in offensive ones.[8]

Among all these additional features, only the hashtag-related features did not improve our original results based on character and POS *n*-grams. We will present detailed results in Section 4.

3.3 Methodology

We used the machine learning library scikit-learn (v0.20.1) (Pedregosa et al., 2011) for Python (v3.7.1) and selected the Support Vector Classifier as our model. We achieved best results on our development data with a linear model default settings.[9] Additional experiments with the Random Forest Classifier, including grid search for parameter tuning, did not yield better results.

3.4 Evaluation

For evaluation, we used the scorer provided by the shared task.[10] It reports accuracy as percent correct, precision and recall for each subset (OFFENSE, OTHER), and the macro-averaged F1-score which is the harmonic mean of the results of the two subsets.

mean=1.69; OTHER: median=1, mean=1.52; Wilcoxon rank sum test: W=1889000, p<0.001.

[8] Period frequency: OFFENSIVE: median=0, mean=0.94; OTHER: median=1, mean=1.08; Wilcoxon rank sum test: W=1574100, p<0.001.

[9] Tests with the constructor option 'probability =True' yielded identical results with slower performance. For details see `https://scikit-learn.org/stable/modules/svm.html#svm-kernels`.

[10] `https://projects.fzai.h-da.de/iggsa/evaluation-tool/`

	Accuracy			OFFENSE			OTHER			Average		
	Perc.	corr.	total	P	R	F	P	R	F	P	R	F
dev	76.94	307	399	84.62	45.21	58.93	75.08	95.26	83.97	79.85	70.23	71.45
test	72.58	2,200	3,031	64.76	31.44	42.33	74.02	91.95	82.02	69.39	61.69	**65.32**

Table 2: Results on the development set and on the final test set (Average= macro-averaged score, Perc.= percent correct, corr.= number of correct tweets, total= number of all tweets, P=precision, R=recall, F= F1-score).

System	Accuracy		OFFENSE			OTHER			Average		
	Perc.	corr.	P	R	F	P	R	F	P	R	F
char (= baseline)	76.19	304	84.00	43.15	57.01	74.38	95.26	83.54	79.19	69.20	70.27
char #/tw	75.94	303	82.89	43.15	56.76	74.30	94.86	83.33	78.60	69.21	70.05
char #-length	76.19	304	84.00	43.15	57.01	74.38	95.26	83.54	79.19	69.20	70.27
char POS	76.44	305	84.21	43.84	57.66	74.61	95.26	83.68	79.41	69.55	70.67
char ./tw	76.44	305	84.21	43.84	57.66	74.61	95.26	83.68	79.41	69.55	70.67
char !/tw	76.44	305	83.33	44.52	58.04	74.77	94.86	83.62	79.05	69.69	70.83
char pu-seq/tw	76.44	305	83.33	44.52	58.04	74.77	94.86	83.62	79.05	69.69	70.83
char @/tw	76.44	305	83.33	44.52	58.04	74.77	94.86	83.62	79.05	69.69	70.83
char ,/tw	76.69	306	84.42	44.52	58.30	74.84	95.26	83.83	79.63	69.89	71.06
char ?/tw	76.69	306	84.42	44.52	58.30	74.84	95.26	83.83	79.63	69.89	71.06
char w/tw	**76.94**	307	84.62	45.21	**58.93**	75.08	95.26	**83.97**	79.85	70.23	**71.45**
char w/tw ,/tw	**76.94**	307	84.62	45.21	**58.93**	75.08	95.26	**83.97**	79.85	70.23	**71.45**
char POS w/tw	**76.94**	307	84.62	45.21	**58.93**	75.08	95.26	**83.97**	79.85	70.23	**71.45**
char ,/tw ?/tw	76.69	306	84.42	44.52	58.30	74.84	95.26	83.83	79.63	69.89	71.06
char w/tw @/tw	76.69	306	83.54	45.21	58.67	75.00	94.86	83.77	79.27	70.03	71.22
char POS w/tw ./tw	75.94	303	82.05	43.84	57.14	74.45	94.47	83.28	78.25	69.15	70.21
char POS w/tw @/tw	76.44	305	83.33	44.52	58.04	74.77	94.86	83.62	79.05	69.69	70.83
all features	74.94	299	80.26	41.78	54.96	73.68	94.07	82.64	76.97	67.93	68.80

Table 3: Results of the ablation study and a model with all features (on the development set, n=399); best results in bold face (char=character n-grams, pu-seq=punctuation sequence, w= tokens, /tw= per tweet, Average= macro-averaged).

Averaging this way makes sure that in unbalanced settings, in which one subset is much larger than the other, the results on the larger subset do not obliterate the results on the smaller one. In case of the training and development data, the subset OTHER was much larger than the subset OFFENSE, cf. Table 1.

We optimized our system for the macro-averaged F1-score on our development set, since this score was the official ranking function in the shared task.

4 Results

We submitted one set of results (based on one run), obtained by one of the systems that had the best results on our development set: a linear SVM using character n-grams, POS n-grams, and tweet length as features (= system *char POS w/tw* in Table 3).

4.1 Official Shared Task Results

Our best system on the development set achieved a macro-averaged F1-score of 65.32 on the shared task's test data, see Table 2.

Overall, the system did not generalize well to the final test data. We observe a loss of about 6 points in macro-averaged F1-score from 71.45 on the development data to 65.32 on the test data. The main decrease is due to a loss of about 20 points in precision on the OFFENSE class, followed by about 14 points in recall. The effect on the OTHER class is much smaller with only about 3 points loss in the F1-score from 83.97 to 82.02.

4.2 Ablation Study

We tested the best bag-of-words setting (character n-grams of size 5-10 and POS n-grams of size 3-9) with different additional feature combinations. Table 3 shows the results including some identical performances for the sake of completeness. The version with all features includes the following features in addition to the bag-of-words features (all measured per tweet): *number of tokens* (w/tw), *number of @'s* (@/tw), *number of #'s* (#/tw), *max_length of hashtag* (#-length), *mean_length of hashtags*, *number of commas* (,/tw), *number of periods* (. /tw), *number of exclamation marks* (!/tw), *number question marks* (?/tw), *punctuation sequences longer than one* (pu-seq/tw).

Using only character *n*-grams provides a solid baseline (*char*). Adding the number of hashtags per tweet decreased the results slightly (*char #/tw*). Adding other individual features increased the overall result. POS *n*-grams improved precision in both classes marginally (*char POS*). The three best combinations used the number of words per tweet (*char w/tw*) as a feature. However, they outperformed models with other feature combinations only slightly. The difference was mostly due to improved recall of offensive tweets (of about 1.4 points in comparison to the model based on characters and POS *n*-grams only). We submitted only the best system combining character information with tweet length and POS information (*char POS w/tw*), hoping that POS *n*-grams can capture some generalizations. It differed in the annotation of the dev set in two tweets from the other two best performing systems (all three systems got these tweets wrong). The other two best performing variants (*char w/tw* and *char w/tw , /tw*) yielded the exact same annotation results. The comma frequency (*, /tw*) does not seem to add additional information which is not also contained in tweet length.

The full version performed systematically worse than systems with fewer features. This could be an effect of overfitting.

We also tested adding unicode strings of emojis as *n*-gram features which did not effect the results in a positive way (not documented in Table 3).

In addition, we experimented by replacing all @-strings with a placeholder *NE* on the token level by keeping the original length information. The idea was to reduce overfitting on individual users. The results on our dev set were discouraging (macro-avgeraged F1-score using *char POS w/tw*: 67.46), so we did not further pursue this approach.

5 Conclusion and Future Work

By participating in the GermEval Shared Task in the setting of a compact introductory course we learned how to conduct the basic steps that are necessary when approaching a machine learning problem: From choosing a model to setting the parameters to extracting features from the data set and implementing them in the algorithm. For our final classifier, we used an SVM algorithm and optimized the system using several features of which character *n*-grams and the length of the tweets with and without POS *n*-grams proved to be most effective.

We obtained an F-score of 65.32 on the test data. In future experiments, the score could possibly be improved by a larger training set and selecting more elaborate features: Montani and Schüller (2018), for example, obtained a high F-score in the GermEval 2018 Shared Task on the Identification of Offensive Language making use of the TF/IDF scores of token *n*-grams. They calculated the TF/IDF for each *n*-gram within each class (i.e. OFFENSE and OTHER) and created a feature that contained only those TF/IDF scores with a document frequency within a certain range (determined by a grid search). Thereby, they reduced the token *n*-gram counts to only those *n*-grams that are important for one of the classes.

Another promising source for potential feature extraction could be the emojis in the tweets with which we only did preliminary tests. They are an important characteristic of twitter data and reveal valuable information about the author's intentions or emotional state. A semantic annotation (i.e. positive, negative) of each emoji type, perhaps with the help of the emoji descriptions in the unicode table, would precede the feature creation. This annotation would have to be done partly manually because emojis can be represented not only by simple but also by complex unicode containing variation selectors.

References

Melanie Andresen, Melitta Gillmann, Jowita Grala, Sarah Jablotschkin, Lea Röseler, Eleonore Schmitt, Lena Schnee, Katarina Straka, Michael Vauth, Sandra Kübler, and Heike Zinsmeister. 2019. The HUIU Contribution to the GermEval 2019 Shared Task 1. In *Proceedings of the GermEval 2019 Workshop*, Erlangen, Germany.

Thorsten Brants, 1998. *TnT–A Statistical Part-of-Speech Tagger.* Universität des Saarlandes, Computational Linguistics, Saarbrücken, Germany.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *arXiv e-prints*, page arXiv:1810.04805, Oct.

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, Minneapolis, MN.

Joaquín Padilla Montani and Peter Schüller. 2018. TUWienKBS at GermEval 2018: German abusive tweet detection. In *Proceedings of the GermEval 2018 Workshop*, pages 45–50, Vienna, Austria. Austrian Academy of Sciences.

Olutobi Owoputi, Brendan O'Connor, Chris Dyer, Kevin Gimpel, Nathan Schneider, and Noah Smith. 2013. Improved part-of-speech tagging for online conversational text with word clusters. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 380–390, Atlanta, GA.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Anne Schiller, Simone Teufel, Christine Stöckert, and Christine Thielen. 1999. Guidelines für das Tagging deutscher Textcorpora mit STTS. Technical report, Institut für maschinelle Sprachverarbeitung & Seminar für Sprachwissenschaft, Stuttgart & Tübingen, Germany.

Heike Telljohann, Erhard W. Hinrichs, Sandra Kübler, and Heike Zinsmeister, 2006. *Stylebook for the Tübingen Treebank of Written German (TüBa-D/Z)*. Seminar für Sprachwissenschaft, Universität Tübingen, Germany.

Michael Wiegand, Melanie Siegel, and Josef Ruppenhofer. 2018. Overview of the GermEval 2018 Shared Task on the Identification of Offensive Language. In *Proceedings of the GermEval 2018 Workshop*, pages 1–10, Vienna, Austria.

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 13th International Workshop on Semantic Evaluation*, pages 75–86, Minneapolis, MN.