RuG at GermEval: Detecting Offensive Speech in German Social Media

Xiaoyu Bai[*], **Flavio Merenda**[*∓], **Claudia Zaghi**[*], **Tommaso Caselli**[*], **Malvina Nissim**[*]
[*] Rikjuniversiteit Groningen, Groningen, The Netherlands
[∓] Università degli Studi di Salerno, Salerno, Italy
f.merenda|t.caselli|m.nissim@rug.nl x.bai.5|c.zaghi@student.rug.nl

Abstract

This paper reports on the systems the RuG Team submitted to the GermEval 2018 - Shared Task on the Identification of Offensive Language in German tweets. We submitted three systems to Task 1, targeting the problem as a binary classification task, and only one system for Task 2, addressing a fine-grained classification of offensive tweets in different categories. Preliminary evaluation of the systems has been conducted on a fixed validation set from the training data. The best macro-F1 score for Task 1, binary classification, is 75.45, obtained by an ensemble model composed by a Linear SVM, a CNN, and a Logistic Regressor as a meta-classifier. As for Task 2, multi-class classification, we obtained a macro-F1 of 40.75 using a multi-class Linear SVM.

1 Introduction

The spread of Social Media, and especially of micro-blog platforms such as Facebook and Twitter, has been accompanied by a growth in on-line **hate speech**. Several countries, including the EU, use this expression as a legal term. For instance, the EU Council Framework Decision 2008/913/JHA[1] specifically defines hate speech as "the public incitement to violence or hatred directed to groups or individuals on the basis of certain characteristics, including race, colour, religion, descent and national or ethnic origin". In this work, following (Schmidt and Wiegand, 2017), hate speech is used as an umbrella term to cover a variety of user-generated content phenomena, such as abusive or hostile messages (Nobata et al., 2016), offensive language, cyberbullying (Reynolds et al., 2011; Xu et al., 2012; Zhong et al., 2016), profanity, insults, toxic conversations (Wulczyn et al., 2017), among others.

Although the EU code of conduct on illegal on-line hate speech forces companies to actively remove hate speech messages in their platforms, the phenomenon is so widespread that ways for the automatic classification of on-line content are advocated and necessary (Bleich, 2014; Nobata et al., 2016; Kennedy et al., 2017). The growing interest in this topic is also shown by recent dedicated workshops (e.g. the Abusive Language Workshop (AWL)[2], now at its second edition), datasets in English and other languages[3], and evaluation exercises, such as the Hate Speech Detection task[4] at the EVALITA 2018 Evaluation Campaign for Italian.

The GermEval 2018 - Shared Task focuses on the automatic identification of *offensive language* in German tweets. In the task setting, *offensive language* is defined as "hurtful, derogatory or obscene comments made by one person to another person". The task is organized into two sub-tasks: i.) Task 1, formulated as a binary classification problem, where each tweet has to be classified either as `OFFENSIVE` or as `OTHER`; and ii.) Task 2, formulated as multi-class classification problem, addressing a fine-grained distinction of the offensive tweets, labeled as `INSULT`, `ABUSE`, and `PROFANITY`, as well as the `OTHER` category. According to the Annotation Guidelines (Ruppenhofer et al., 2018), the `OTHER` category is defined as any utterance either having a positive or neutral polarity, or having a negative polarity but not expressing any of the target categories of `INSULT`, `ABUSE`, and `PROFANITY`. Notice also that the category

[1] https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=LEGISSUM:l33178

[2] https://sites.google.com/view/alw2018

[3] https://sites.google.com/view/alw2018/resources?authuser=0

[4] http://di.unito.it/haspeedeevalita18

PROFANITY is used to mark utterances that express non-acceptable language (e.g. swearwords) without targeting (an) individual(s), thus basically not expressing hate speech.

This paper illustrates the settings of our participating systems. Although we mainly focused on Task 1, to which we submitted three different runs, we also participated to Task 2 with only one run. Code and outputs are publicly available [5]. In the remainder of the paper, we first discuss some of the resources we used, including additional publicly available data we obtained (Section 2), then describe each of our submitted system runs, including their results on a validation set (Section 3 and Section 4). We also present a discussion on what we tried but did not work during system development (Section 5). We then conclude with a quick overview of previous works in this topic (Section 6) and reflections on future directions (Section 7).

2 Data and Resources

All of our runs, both for Task 1 and for Task 2, are based on supervised approaches, where data (and features) play a major role for the final results of a system. This section illustrates the datasets and language resources used in the final submissions.

2.1 Resources Provided by Organizers

We have been provided with 5009 labeled German tweets as training data. Table 1 illustrates the distribution of the classes for each of the subtask.

Class	**Samples**
Task 1: Binary task	
OFFENSE	1,688
OTHER	3,321
Task 2: Multi-class task	
ABUSE	1,022
INSULT	595
PROFANITY	71
OTHER	3,321

Table 1: Class distribution in the share task training data for Task 1 and Task 2.

We also experimented with the following resources made available by the organizers:

- German word embeddings pre-trained on either Twitter or Wikipedia data (Cieliebak et al., 2017; Deriu et al., 2017) available from SpinningBytes[6]. Embeddings of sizes 200, 100 and 52 dimensions are available. We used the 52 dimension embeddings.
- A comprehensive list of offensive words in German, obtained from the website `http://www.hyperhero.com/de/insults.htm`.

2.2 Additional Resources

Source-driven Embeddings A major focus of our contribution is the development of offense-rich, or highly polarized, word embedding representations. To build them, we scraped data from social media communities on Facebook pages. The working hypothesis, grounded on previous studies on on-line communities (Pariser, 2011; Bozdag and van den Hoven, 2015; Seargeant and Tagg, 2018), is that each on-line community represents a different source of data, and consequently, their user-generated contents can be used as proxies for specialized information. We thus acquired *source-driven embeddings* by extracting publicly available comments from a set of German-language Facebook communities that are likely to contain offensive language, and induce word embeddings on the data extracted. The idea is that the embeddings obtained in this manner will be more sensitive to offensive language, with similarly offensive terms being placed closer to each other in the vector space. Table 2 shows the Facebook pages we used (which largely relate to right-wing populist political groups) and the respective number of comments we extracted from each page.

Page Name	**Comments**
AfD-Fraktion AGH	6,933
Alice Weidel	279,435
Asylflut stoppen	3,461
NPD - Die soziale heimatpartei	138,611
Total	**428,440**

Table 2: List of public Facebook pages from which we obtained comments and number of extracted comments per page.

The embeddings were randomly initialized and generated with the word2vec skip-gram model (Mikolov et al., 2013), using a context window of 5, and minimum frequency 1. The final vocabulary amounts to 313,443 words. These embeddings, referred to as "hate embeddings" here-

[5] https://github.com/malvinanissim/germeval-rug

[6] https://www.spinningbytes.com/resources/wordembeddings/

after, were induced as vectors of 300 dimensions in one setting and of 52 dimensions in another.

We also trained 52 dimensional word embeddings on the shared task training data, using our 52 dimension hate embeddings to initialize the process instead of random initialization. We refer to this further set of embeddings as "hate-oriented embeddings".

To summarize, we generated three sets of word embeddings:

- 300 dimension *hate embeddings* based on Facebook comments;
- 52 dimension *hate embeddings* based on Facebook comments;
- 52 dimension *hate-oriented embeddings*, that incorporate information from the hate embeddings plus the shared task training data.

Extra Training Data Given the dimension of the training data, and especially the lower number of "offensive" tweets, we found an additional dataset of social media messages annotated for offensive language and hate speech, the Political Speech Project (Bröckling et al., 2018). The dataset is part of a journalistic initiative to chart the quality of on-line political discourse in the EU. Almost 40 thousands Facebook comments and tweets between February 21 and March 21, 2018, were collected and manually annotated by an international team of journalists from four countries (France, Italy, Germany, and Switzerland) for level and category of offense. Out of a total of 9,861 utterances from Germany, we extracted and used as extra-training data 549 utterances that were labeled as offensive. We will refer to this extra dataset henceforth as PSP data.

3 Our Submissions

We detail in this section our final submissions to the task, three of which address Task 1, binary classification, and one Task 2, multi-class classification.

3.1 Submission 1: Binary Model with SVM

Our first submission, named `rug_coarse_1.txt`, contains the predictions for the binary task made by an SVM model using various linguistic features.[7] The system was implemented using the Scikit-Learn Python toolkit (Pedregosa et al., 2011).

We performed minimal pre-processing by:

- replacing all mentions/usernames with the generic form *User*;
- removing the line break characters |*LBR*|;
- removing the hash character from all hashtags;
- removing stop words using the Python module `stop-words`[8]

We used two groups of surface features, namely: i.) unigrams and bigrams; and ii.) character n-grams in the range between 3 and 7.

The resulting sparse vector representation of each (training) sample is concatenated with its dense vector representation. The dense vector representation for each tweet is obtained as follows: for every word w in a tweet t, we derived a 52 dimension representation, $\vec{w}$, by means of a look-up in the 52 dimension hate-oriented embeddings. We then performed max pooling over all these word embeddings, $\vec{w}$, to obtain a 52 dimension embedding representation of the full tweet, $\vec{t}$. Words not covered in the hate-oriented embeddings were ignored.

The classifier is a linear SVM with unbalanced class weights. Since the training data is unbalanced and the class `OFFENSE` under-represented, we chose to specify the SVM class weights for `OTHER` and `OFFENSE` as 1 and 3, respectively. We used default values for the other hyper-parameters.

3.2 Submission 2: Binary Model with CNN

Our second submission, `rug_coarse_2.txt`, is based on a Convolutional Neural Network (CNN) architecture for sentence classification (Kim, 2014; Zhang and Wallace, 2015) using Keras (Chollet and others, 2015). The architecture of the model is composed of the following layers:

- A word embeddings input layer using the 300 dimension hate word embeddings (see 2.2);
- A convolution layer;
- A max-pooling layer;
- A fully-connected layer;
- A sigmoid output layer.

[7] In all of our submissions we use the string `XXX` as the dummy label for the task not worked on.

[8] `https://pypi.org/project/stop-words/`

This is a simple architecture with one convolutional layer built on top of a word embedding layer. The embedding layer output corresponds to a tensor of shape three: instances, sequence length and embedding dimension. Later, this output is connected to the convolution layer.

The max-pooling layer output is flattened, concatenated, and fed to the fully-connected layer composed of of 50 hidden-units with the ReLU activation function. The final output layer with the sigmoid activation function computes the probabilistic distribution over the two labels (other network hyperparameters: `Number of filters`: 6; `Filter sizes`: 3, 5, 8; `Strides`: 1; `Activation function`: Rectifier; `Padding`: valid). For our model we chose the binary cross-entropy loss function. As optimization function we employed the Adaptive Moment Estimation (Adam). To train our system, we set a batch size of 64 and we ran it for 10 epochs. To reduce risks of overfitting, we applied two dropout values, 0.6 and 0.8 We added the first dropout layer between the embeddings and the convolution layer, and the second one between the max-pooling and the fully-concatenated layer.

Finally, for this system, the original training data was extended with the 549 PSP data labeled as offensive, thus yielding a new class distribution as shown in Table 3.

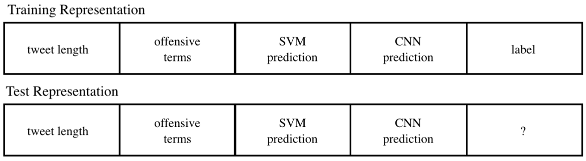
Class	**Samples**
OFFENSE	2,237
OTHER	3,321
Total	**5,558**

Table 3: Class distribution in the training data extended with PSP

3.3 Submission 3: Binary Ensemble Model

Our third submission, named `rug_coarse_3.txt`, is an ensemble model that combines the SVM and CNN models described in Submissions 1 and 2 (Sections 3.1 and 3.2) and a meta-classifier based on a Logistic Regressor classifier.

Each message is composed by 2 groups of surface features, namely, the length of the tweet in terms of number of characters (tweet length), and the number of times an offensive term from the above-mentioned list of offensive German terms (Section 2.1) occurs in the tweet, normalized by the tweet's length (offensive terms), plus the predictions from the Linear SVM and the CNN models. Figure 1 graphically illustrates the representation of each message. The top part illustrates a training sample, while the bottom part a test sample. Such representations are fed as features to the Logistic Regressor, implemented using Scikit-Learn using the default parameters.[9]



Training Representation

tweet length	offensive terms	SVM prediction	CNN prediction	label

Test Representation

tweet length	offensive terms	SVM prediction	CNN prediction	?

Figure 1: Feature representation of each sample fed to the ensemble model. On top, the representation of a training sample, on bottom, the representation of a test sample.

The predictions outputted by the SVM are in the form of the complementary probabilities for either of the two classes, those by the CNN are in form of the probability of the class OFFENSE. The predictions of the SVM and the CNN for the 5009 training samples which we need to feed to the meta-classifier at training time were obtained via 5-fold cross validation. At test time, each system was trained on the full training dataset and produced predictions for each of the test samples, which are then fed as features to the meta-classifier.

Notice that, as described in the previous sections, the CNN was trained on a dataset which featured the addition of the PSP data, while the SVM did not, as this did not prove useful at development time (see Section 5). Thus, in the case of the CNN system, 5-fold cross validation in fact yielded predictions for each of the 5009 training samples, plus the 549 added samples from the PSP data, which were then discarded when training the meta-classifier.

3.4 Submission 4: Multi-Class with SVM

The file named `rug_fine_1.txt` is our only submission to the fine-grained/multi-class task (Task 2), containing predictions by an SVM model. The system and features used are identical to those used in Submission 1 (Section 3.1), except that the SVM class weights for the four classes OTHER, ABUSE, INSULT and PROFANITY were set as 0.5, 3, 3 and 4, respectively. PROFANITY was

[9] http://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html

given the highest weight since it is a severely under-represented class.

4 Preliminary Results

Table 4 gives an overview of the preliminary results of our systems in terms of accuracy and macro-F1 score. The systems' results are also compared against two naive baseline models based on the majority class (i.e. OTHER). All scores were obtained by training on 80% of the 5009-sample training data and testing on a fixed development set of 20%.

	Accuracy	**F1 (macro)**
Task 1: Binary task		
Baseline	*65.27*	*39.49*
SVM binary	76.25	71.90
CNN binary	76.85	73.05
Ensemble binary	78.34	74.45
Task 2: Multi-class task		
Baseline	*65.27*	*19.75*
SVM multi-class	71.66	40.75

Table 4: Results of our submitted systems and majority-class baselines in terms of accuracy and macro-average F1 training on 80% of the training set provided, and testing on the remaining 20%.

5 Methods Not Adopted

When developing our system we experimented with a series of additions and variations aimed at improving performance. Not everything worked or made a difference either using cross-validation or randomly picked development sets, but we deem it interesting to report on such attempts in this paper.

Data Given the significant under-representation of the classes INSULT and PROFANITY in the multi-class setting, we experimented with upsampling them by duplicating the samples from these two classes. However, this did not yield any gains in performance. With respect to the additional PSP dataset, we found that unlike the CNN, the SVM did not benefit from the addition of the 549 additional offensive samples and therefore did not adopt this for the final submissions. Moreover, we also experimented with the extension of the training data with *all* samples from the PSP dataset (9,312 neutral/other, 549 offensive), instead of only adding the 549 samples annotated as offensive. However, both the CNN and the SVM suffered from this, likely due to the resulting inflation of the class OTHER.

Representations For the SVM we experimented with different sets of word embeddings which were used to obtain dense-vector representations of full samples in the manner described in Section 3.1. The 52 dimension Twitter and Wikipedia embeddings from SpinningBytes performed similarly. Furthermore, we also tried to join them by concatenating their representations for each word and tested different methods of dealing with the words that are covered by one set of embeddings only. In one setting, we left the embeddings of these words unchanged and used Principle Component Analysis to reduce the dimensions of all other word vectors back to 52. Thus, all embeddings were of 52 dimensions, but those words covered by both sets of embeddings incorporated distributional information from both Twitter and Wikipedia in their representations. In another setting, we obtained unreduced, concatenated embeddings of 104 dimensions, using padding for words which only occur in either the Twitter or the Wikipedia embeddings. Our experiments showed, however, that these alternative word embeddings performed worse than those we used in our final submissions.

Algorithms In the ensemble system we also experimented with using another Linear SVM as the meta-classifier. However, its performance in this capacity was inferior to that of our final choice, i.e. a Logistic Regressor.

6 Related Work

Several models have been presented in the literature to detect hate speech and its related concepts (offensive language, cyberbullying and profanity among others).

The task has been mainly addressed by means of rule-based methods or supervised classifiers. Rule-based methods (De Marneffe and Manning, 2008; Mondal et al., 2017; Pelosi et al., 2017; Xu and Zhu, 2010; Su et al., 2017; Palmer et al., 2017) heavily rely on lexical resources such as dictionaries, thesauri, sentiment lexicons, as well as syntactic patterns and POS relations.

Supervised approaches have shown to obtain good results, although they suffer from limitations as far as the size and domain of the training data is concerned. Support Vector Machine and Convolutional Neural Network classifiers turned out to be efficient algorithms for this task. Simple SVM models with word embeddings (Del Vigna et al., 2017) and TF-IDF n-grams (Davidson et al., 2017) showed competitive performances. On the other hand, CNN architectures are initialized with word embeddings that can be obtained "on the fly" using the training data or from some pre-trained representations (Badjatiya et al., 2017; Gambäck and Sikdar, 2017; Park and Fung, 2017; Badjatiya et al., 2017). Other classifiers widely employed in literature are LSTMs (Del Vigna et al., 2017; Badjatiya et al., 2017; Gao and Huang, 2017; Chu et al., 2016), and Logistic Regressors (Djuric et al., 2015; Davidson et al., 2017; Gao and Huang, 2017).

A remarkable experiment developed an ensemble classifier combining the predictions of a logistic regression model with the ones obtained with an LSTM neural network (Gao and Huang, 2017).

7 Conclusions and Future Work

This paper reports on the RuG Team submissions to Task 1 and 2 of the GermEval 2018 - Shared Task on the Identification of Offensive Language. Our team focused mainly on Task 1, a binary classification task aiming at classifying German tweets either as `OFFENSIVE` or `OTHER`. In the development of our systems, we put our efforts on the development of embedding representations that could reduce the dependence of the models on the training data, exploiting Facebook on-line communities to generate such data (source-based embeddings). The results on a fixed validation set composed by 20% of the training data have shown that the use of these "hate embeddings" is beneficial. Of the three systems we submitted for Task 1 (a linear SVM, a CNN, and an ensemble model based on the SVM and CNN predictions and extended with basic surface features), the ensemble model obtains the best results (macro-F1 74.45), followed by the CNN (macro-F1 73.05), and, finally, the SVM (macro-F1 71.90).

Task 2, fine-grained classification, was addressed with a simple Linear SVM, using as features word and characters n-grams. The fine-grained classification proved harder than the binary one, also for the limited amount of the training data. The system has a macro-F1 of 40.75 on the same validation set as the binary task.

We are planning to conduct a deep error analysis once the official scores and gold test data will be made available, so as to have a better understanding of the limitations of our models. Furthermore, we also plan to extend the source-based approach to collect polarized embeddings and to test it on other languages as well.

Acknowledgments

The authors want to thank Angelo Basile for his feedback in the early stages of this work. A special thank goes to Rania Wazir for her help in obtaining the PSP data.

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*, pages 759–760. International World Wide Web Conferences Steering Committee.

Erik Bleich. 2014. Freedom of expression versus racist hate speech: Explaining differences between high court regulations in the usa and europe. *Journal of Ethnic and Migration Studies*, 40(2):283–300.

Engin Bozdag and Jeroen van den Hoven. 2015. Breaking the filter bubble: democracy and design. *Ethics and Information Technology*, 17(4):249–265.

Marie Bröckling, Vincent Coquaz, Alexander Fanta, Alison Langley, Mauro Munafò, Julian Pütz, Francesca Sironi, Leo Thüer, and Rania Wazir. 2018. Political Speech Project. https://rania.shinyapps.io/PoliticalSpeechProject/, May.

François Chollet et al. 2015. Keras. `https://github.com/fchollet/keras`.

Theodora Chu, Kylie Jue, and Max Wang. 2016. Comment abuse classification with deep learning. Von https://web. stanford. edu/class/cs224n/reports/2762092. pdf abgerufen.

Mark Cieliebak, Jan Milan Deriu, Dominic Egger, and Fatih Uzdilli. 2017. A twitter corpus and benchmark resources for german sentiment analysis. In *5th International Workshop on Natural Language Processing for Social Media, Boston, MA, USA, December 11, 2017*, pages 45–51. Association for Computational Linguistics.

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

Marie-Catherine De Marneffe and Christopher D Manning. 2008. Stanford typed dependencies manual. Technical report, Technical report, Stanford University.

Fabio Del Vigna, Andrea Cimino, Felice DellOrletta, Marinella Petrocchi, and Maurizio Tesconi. 2017. Hate me, hate me not: Hate speech detection on facebook. *In Proceedings of the First Italian Conference on Cybersecurity (ITASEC17), Venice, Italy.*

Jan Deriu, Aurelien Lucchi, Valeria De Luca, Aliaksei Severyn, Simon Müller, Mark Cieliebak, Thomas Hofmann, and Martin Jaggi. 2017. Leveraging large amounts of weakly supervised data for multi-language sentiment classification. In *Proceedings of the 26th international conference on world wide web*, pages 1045–1052. International World Wide Web Conferences Steering Committee.

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.

Björn Gambäck and Utpal Kumar Sikdar. 2017. Using convolutional neural networks to classify hate-speech. In *Proceedings of the First Workshop on Abusive Language Online*, pages 85–90.

Lei Gao and Ruihong Huang. 2017. Detecting online hate speech using context aware models. *arXiv preprint arXiv:1710.07395*.

George Kennedy, Andrew McCollough, Edward Dixon, Alexei Bastidas, John Ryan, Chris Loo, and Saurav Sahay. 2017. Technology solutions to combat online harassment. In *Proceedings of the First Workshop on Abusive Language Online*, pages 73–77.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882*.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Mainack Mondal, Leandro Araujo Silva, and Fabricio Benevenuto. 2017. A measurement study of hate speech in social media. In *Proceedings of the 28th ACM Conference on Hypertext and Social Media*, pages 85–94. ACM.

Chikashi Nobata, Joel Tetreault, Achint Thomas, Yashar Mehdad, and Yi Chang. 2016. Abusive language detection in online user content. In *Proceedings of the 25th International Conference on World Wide Web*, pages 145–153. International World Wide Web Conferences Steering Committee.

Alexis Palmer, Melissa Robinson, and Kristy K Phillips. 2017. Illegal is not a noun: Linguistic form for detection of pejorative nominalizations. In *Proceedings of the First Workshop on Abusive Language Online*, pages 91–100.

Eli Pariser. 2011. *The filter bubble: What the Internet is hiding from you*. Penguin UK.

Ji Ho Park and Pascale Fung. 2017. One-step and two-step classification for abusive language detection on twitter. *arXiv preprint arXiv:1706.01206*.

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(Oct):2825–2830.

Serena Pelosi, Alessandro Maisto, Pierluigi Vitale, and Simonetta Vietri. 2017. Mining offensive language on social media. In *CLiC-it*.

Kelly Reynolds, April Kontostathis, and Lynne Edwards. 2011. Using machine learning to detect cyberbullying. In *Machine learning and applications and workshops (ICMLA), 2011 10th International Conference on*, volume 2, pages 241–244. IEEE.

Josef Ruppenhofer, Melanie Siegel, and Michael Wiegand. 2018. Guidelines for IGGSA Shared Task on the Identification of Offensive Language. http://www.coli.uni-saarland.de/ miwieg/Germeval/guidelines-iggsa-shared.pdf, March.

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. Association for Computational Linguistics, Valencia, Spain*, pages 1–10.

Philip Seargeant and Caroline Tagg. 2018. Social media and the future of open debate: A user-oriented approach to Facebook's filter bubble conundrum. *Discourse, Context & Media*.

Hui-Po Su, Zhen-Jie Huang, Hao-Tsung Chang, and Chuan-Jie Lin. 2017. Rephrasing profanity in chinese text. In *Proceedings of the First Workshop on Abusive Language Online*, pages 18–24.

Ellery Wulczyn, Nithum Thain, and Lucas Dixon. 2017. Ex machina: Personal attacks seen at scale. In *Proceedings of the 26th International Conference on World Wide Web*, pages 1391–1399. International World Wide Web Conferences Steering Committee.

Zhi Xu and Sencun Zhu. 2010. Filtering offensive language in online communities using grammatical relations. In *Proceedings of the Seventh Annual Collaboration, Electronic Messaging, Anti-Abuse and Spam Conference*, pages 1–10.

Jun-Ming Xu, Kwang-Sung Jun, Xiaojin Zhu, and Amy Bellmore. 2012. Learning from bullying traces in social media. In *Proceedings of the 2012 conference of the North American chapter of the association for computational linguistics: Human language technologies*, pages 656–666. Association for Computational Linguistics.

Ye Zhang and Byron Wallace. 2015. A sensitivity analysis of (and practitioners' guide to) convolutional neural networks for sentence classification. *arXiv preprint arXiv:1510.03820*.

Haoti Zhong, Hao Li, Anna Cinzia Squicciarini, Sarah Michele Rajtmajer, Christopher Griffin, David J Miller, and Cornelia Caragea. 2016. Content-driven detection of cyberbullying on the instagram social network. In *IJCAI*, pages 3952–3958.