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

German Hatespeech classification with Naive Bayes and Logistic Regression - hshl at GermEval 2019 - Task 2

Kristian Rother
Hochschule Hamm-Lippstadt
Marker Allee 76-78
59063 Hamm
kristian.rother@hshl.de

Achim Rettberg
Hochschule Hamm-Lippstadt
Marker Allee 76-78
59063 Hamm
achim.rettberg@hshl.de

Abstract

This paper describes the entries *hshl_coarse_1* and *hshl_fine_1* for Subtask I (Binary Classification) and Subtask II (Fine-grained classification) of Task 2 of the the GermEval 2019 competition. For this task, German tweets were classified as either OFFENSE or OTHER (Subtask I) or into the four subcategories PROFANITY, INSULT, ABUSE or OTHER (Subtask II). The entries employ a mixture of character level Logistic Regression and Naive Bayes. The classifiers were trained on the labeled tweets that were provided by the organizers of the shared task. The optimization of the system is outlined in this paper. The system reached an F1-score of 0.7458 and 0.4793 for the two subtasks on the test-set.

1 Introduction

Hate speech is on the rise in online communication and can come in different forms but usually follows certain patterns (Mondal et al., 2017). Additionally social media serves as a breeding ground for deviant behavior following real world incidents (Williams and Burnap, 2015).

Hate speech has psychological consequences for the victims such as fear, anger and vulnerability (Awan and Zempi, 2015) as well as the worry that online threats may become a reality (Awan and Zempi, 2016). Additionally, hate speech can be the harbinger of actual violence. Hate speech towards a group can serve as a predictor of violence towards that group (Müller and Schwarz, 2018a) and Twitter use can fuel hate-crimes (Müller and Schwarz, 2018b).

Institutions and legislators have reacted to this trend towards hate speech. The European Commission and multiple social media companies agreed to a code of conduct on countering illegal hate speech online (European Commission, 2016). Germany passed the *Network Enforcement Act* on September 1st 2017 to enforce fines of up to 50 million Euros against social media companies that fail to delete illegal content (German Bundestag, 2017). The law specifically includes hate speech (§§130, 166 and 185-187 of the Criminal Code).

Due to the negative impact of hate speech and the amount of social media data that is generated every day, automated detection and classification of hate speech has been studied widely. Recent overviews can be found in (Schmidt and Wiegand, 2017) and (Fortuna and Nunes, 2018). However, with some exceptions such as (Ross et al., 2017) and (Van Hee et al., 2015), the scope of the studies is often limited to the English language.

As a result, the GermEval competition was launched in 2018 (Wiegand et al., 2018). By and large, (deep) neural models performed best in 2018, which is why this entry for the 2019 version of the competition focuses on a simpler, classical machine learning approach to provide a contrast to the expected neural models. Therefore, this paper tries to contribute to the improvement of the state of the art in German hate speech detection by describing the entries *hshl_coarse_1* and *hshl_coarse_2* which participated in Taks 2 at GermEval 2019.

2 Experimental Setup

The following section describes the experimental setup, namely all used technical resources, the used data and the chosen architecture.

2.1 Technical Resources

All experiments were conducted in Jupyter Notebooks, version 4.0.2 (Kluyver et al., 2016) running a Python 3.5.0 (Python Software Foundation, 2018) kernel with the following libraries:

- pandas 0.23.4 (McKinney, 2010)
- NumPy 1.11.3 (Oliphant, 2006)

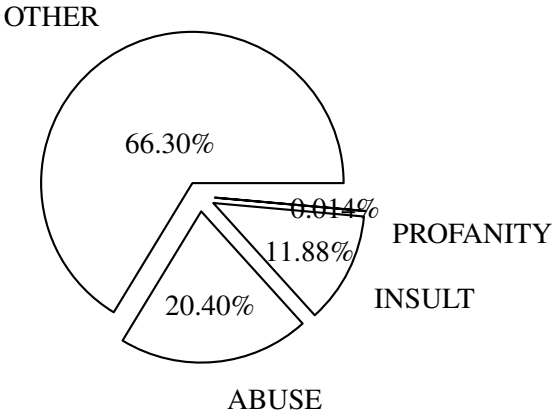
- scikit-learn 0.20 (Pedregosa et al., 2011)
- spaCy 2.0.12 (Honnibal and Montani, 2018)

A fixed seed was used for the random number generators. All models were trained on an end of 2013 MacBook Pro with a 2 GHz Intel Core i7, 8 GB 1600 MHz DDR3 and an Intel Iris Pro 1536 MB GPU[1].

2.2 Data

To train and tune the model, the data from the 2018 competition was used. This data consists of 5009 labeled tweets for the training-set and 3532 labeled tweets for the dev-set[2]. The distribution of the classes for Subtask II is depicted in figure 1. Note that for Subtask I, the classes ABUSE, INSULT and PROFANITY are simply summarized as one class, named OFFENSE and thus this class consists of 33.70% of all tweets. The data is imbalanced. Specifically, the class OTHER makes up two thirds of the data while the class PROFANITY only represents 0.014 percent of all tweets.

Figure 1: Class distribution 2018 training-set.



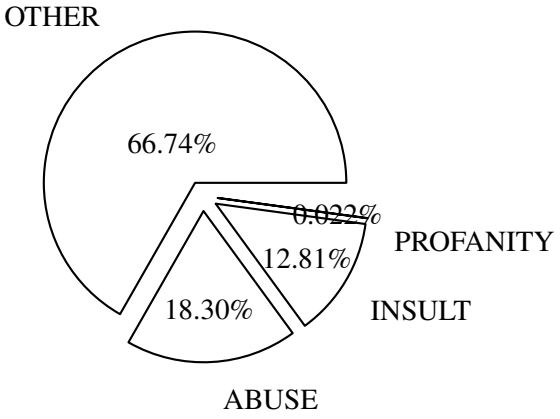
For the final submission, the system was trained on all available labeled tweets. These consist of the 2018 training- and test-set as well as the 2019 training-set for a total of 12.401 labeled tweets[3]. The class distributions for the 2019 training-set are described in figure 2. No additional outside data was used.

[1]With this setup, the entire end-to-end training on the provided development set from 2018 took 14 seconds for Subtask A and 55 seconds for Subtask B.

[2]This dev-set used to be the test-set in the 2018 competition but the organizers released the labels after the competiton was over.

[3]Some tweets were removed because they could not be parsed correctly.

Figure 2: Class distribution 2019 training-set.



2.3 Architecture

A simple bag-of-words approach that relies on the frequency of words to classify the data was used for the competition. Thus, the tweets were vectorized before they were used in the algorithms. For the vectorization, a term-frequency times inverse document-frequency matrix was used. The chosen Logistic Regression model combined with a Naive Bayes approach.

3 Experiments

The following section describes the experiments that were conducted. Because experiments could improve results for one subtask but lead to worse results on the other subtask, it was decided to use Subtask I results to pick the parameters to keep. For all experiments, spaCy was used as the tokenizer with no lemmatization and no stopword list[4]. For the term-frequency matrices, only words that appeared in at least 4 documents were used and words that appeared in more than 40% of documents were ignored. Inverse document-frequency-reweighting (IDFR) and sublinear term frequency scaling were applied, but no smoothing was applied. The parameters are summarized in table 1.

Parameter	Value
Minimum Number of Documents	4
Maximum Frequency of Documents	0.4
IDFR	Yes
Sublinear Term Frequency Scaling	Yes
Smoothing	No

Table 1: Parameters for the term-frequency matrices.

[4]As these didn't improve results in some previous tests.

3.1 Accent Stripping, Casing and Preprocessing

The first experiment was to test unicode accent stripping and lowercasing combinations. These tests were conducted for word-level unigrams. The results are summarized in table 2. Note that the last two cases are the same due to the order of lowercasing and stripping. Notably, removing lowercasing lead to worse results overall while accent stripping provided slightly better results for both subtasks.

Strip	✗	✓	✗	✓
Lowercase	✓	✓	✗	✗
F1 Sub I	0.6655	**0.6665**	0.6467	0.6467
F1 Sub II	0.3970	**0.3975**	0.3917	0.3917

Table 2: Unicode accent stripping and lowercasing. The F1 scores are micro-averaged on the dev-set.

3.2 N-grams

Different word-level n-grams were tried [5]. Neither bigrams nor trigrams outperformed the unigrams for Subtask I. However, bigrams were best for Subtask II. The results are summarized in table 3.

In-word n-grams were also tried but didn't show any improvements.

N-gram	F1 Sub I	F1 Sub II
1,1	**0.6665**	0.3975
1,2	0.6496	**0.4114**
1,3	0.6490	0.4053
2,3	0.5874	0.3090

Table 3: Word-level n-grams. The F1 scores are micro-averaged on the dev-set.

Finally, character-level n-grams with a min to max difference of three were tried. The results are summarized in table 4.

N-gram	F1 Sub I	F1 Sub II
1,4	0.6474	0.3736
2,5	0.6672	0.3807
3,6	**0.6693**	**0.3855**

Table 4: Character-level n-grams. The F1 scores are micro-averaged on the dev-set.

[5] As a note for reproducability, it was discovered after the submission, that this experiment was run with lowercasing set to false eventhough the previous experiment would suggest to use true. However the final results were not impacted by this. And lowercasing was set to true for the following experiments.

The best character-level model outperformed the best word-level model for Subtask I and was thus kept for the next step.

3.3 Hyperparameters of the Logistic Regression

The hyperparameter tuning for the character-level model is summarized below. Automatically adjusting weights inversely proportional to class frequencies (balanced) provided big gains as summarized in table 5.

Balanced	✗	✓
F1 Sub I	0.6693	**0.7142**
F1 Sub II	0.3855	**0.4514**

Table 5: Balanced class weights. The F1 scores are micro-averaged on the dev-set.

Finally, different C-values for the L2 regularization were tested. The results are summarized in table 6. Because C=14 lead to the best score for Subtask I, this value was picked.

C-value	F1 Sub I	F1 Sub II
1.0	0.7142	0.4514
16.0	0.7194	**0.4646**
14.0	**0.7202**	0.4639

Table 6: L2 Regularization. The F1 scores are micro-averaged on the dev-set.

3.4 Task specific text preprocessing

As the last step, Twitter specific preprocessing was applied. The strings *RT* and *'s* and the symbols *:* and *#* were removed. All urls were replaced with xx_url and all @usernames were replaced with xx_username. This increased the F1-score for Subtask I to 0.7228 (and lowered the score for Subtask II to 0.4569).

3.5 Cutoff Value for Subtask I

Another parameter that can be varied is the cutoff value for Subtask I. If the prediction is equal to or more than this cutoff value, the label 'offense' is predicted and otherwise, 'other' is predicted. The logical value to use for the cutoff would be 0.5. However, experiments showed that changing this value also influences the overall result. The best empirical value of 0.485 increased the F1-score for Subtask I to 0.7258.

4 Results

After the submission deadline for the predictions, the organizers calculated various statistics on the test-set. The results for Subtask I are summarized in table 7 and the results for Subtask II are summarized in table 8. The competition was scored on the average F1-score for each task, which is highlighted in bold in the tables.

	Precision	Recall	F1-Score
OFFENSE	0.6800	0.6134	0.6450
OTHER	0.8261	0.8641	0.8447
Average	0.7530	0.7388	**0.7458**
	Accuracy	Correct	Total
	78.39	2376	3031

Table 7: Results for Subtask I on the 2019 test-set. F1 scores are micro-averaged.

	Precision	Recall	F1-Score
ABUSE	0.4312	0.4075	0.4190
INSULT	0.5567	0.2353	03308
OTHER	0.7874	0.9326	0.8538
PROFANITY	0.5000	0.0811	0.1395
Average	0.5688	0.4141	**0.4793**
	Accuracy	Correct	Total
	72.65	2202	3031

Table 8: Results for Subtask II on the 2019 test-set. F1 scores are micro-averaged.

5 Conclusion

The paper presented the submissions *hshl_coarse_1* and *hshl_fine_1* that were entered for the binary and fine-grained hate speech classification task of GermEval 2019. A combination of Naive Bayes and Logistic Regression was used to classify the tweets.

To reach the final model, different experiments on vectorization and preprocessing were run. Additionally, the hyperparameters of the Logistic Regression were tuned. The final model reached micro-averaged F1 scores of 0.7458 and 0.4793 on the test-set for the two subtasks.

All relevant code will be made available at one of the authors' Github repositories[6]. Additionally, lab notes of the different experiments will be added to the repository.

[6]https://github.com/rother/germeval2019

6 Outlook

Further improvements could be made by using an ensemble of different classifiers. Neither the vectorization nor the preprocessing or the hyperparameter search were exhaustive and can be improved upon. Using different tokenizers or other preprocessing strategies could yield better results. Theoretically, the chosen bag of words approach that only relies on word frequencies is inferior to a model that takes word order and more details into account. As such, deep neural models, specifically RNNs, are likely to provide better overall results. Lastly different sampling strategies as a measure against the imbalanced class labels would likely improve the results.

Acknowledgments

We thank the Behr-Hella Thermocontrol GmbH for supporting this research. We also thank all reviewers and the competition organizers.

References

Imran Awan and Irene Zempi. 2015. We fear for our lives: Offline and online experiences of anti-Muslim hostility. *Report,[online] available: http://tellmamauk. org/wp-content/uploads/resources/We% 20Fear% 20For% 20Our% 20Lives. pdf [accessed: 7 January, 2016]*.

Imran Awan and Irene Zempi. 2016. The affinity between online and offline anti-Muslim hate crime: Dynamics and impacts. *Aggression and violent behavior*, 27:1–8.

European Commission. 2016. Code of conduct on countering illegal hate speech online. `http://ec.europa.eu/newsroom/document.cfm?doc_id=42985`.

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.

German Bundestag. 2017. Act to improve enforcement of the law in social networks (network enforcement act). `https://www.bmjv.de/SharedDocs/Gesetzgebungsverfahren/Dokumente/NetzDG_engl.pdf?__blob=publicationFile&v=2`.

Matthew Honnibal and Ines Montani. 2018. spaCy library. `https://spacy.io`.

Thomas Kluyver, Benjamin Ragan-Kelley, Fernando Pérez, Brian Granger, Matthias Bussonnier, Jonathan Frederic, Kyle Kelley, Jessica Hamrick, Jason Grout, Sylvain Corlay, Paul Ivanov, Damián

Avila, Safia Abdalla, and Carol Willing. 2016. Jupyter notebooks – a publishing format for reproducible computational workflows. In F. Loizides and B. Schmidt, editors, *Positioning and Power in Academic Publishing: Players, Agents and Agendas*, pages 87 – 90. IOS Press.

Wes McKinney. 2010. Data structures for statistical computing in python. In *Proceedings of the 9th Python in Science Conference*, volume 445, pages 51–56. Austin, TX.

Mainack Mondal, Leandro Arajo Silva, and Fabrício 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.

Karsten Müller and Carlo Schwarz. 2018a. Fanning the Flames of Hate: Social Media and Hate Crime. CAGE Online Working Paper Series 373, Competitive Advantage in the Global Economy (CAGE).

Karsten Müller and Carlo Schwarz. 2018b. Making America Hate Again? Twitter and Hate Crime under Trump.

Travis E. Oliphant. 2006. *A guide to NumPy*, volume 1. Trelgol Publishing USA.

Fabian Pedregosa, Gal Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, and Vincent Dubourg. 2011. Scikit-learn: Machine learning in Python. *Journal of machine learning research*, 12(Oct):2825–2830.

Python Software Foundation. 2018. Python programming language. `https://python.org`.

Björn Ross, Michael Rist, Guillermo Carbonell, Benjamin Cabrera, Nils Kurowsky, and Michael Wojatzki. 2017. Measuring the reliability of hate speech annotations: The case of the european refugee crisis. *arXiv preprint arXiv:1701.08118*.

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.

Cynthia Van Hee, Els Lefever, Ben Verhoeven, Julie Mennes, Bart Desmet, Guy De Pauw, Walter Daelemans, and Vronique Hoste. 2015. Detection and fine-grained classification of cyberbullying events. In *International Conference Recent Advances in Natural Language Processing (RANLP)*, pages 672–680.

Michael Wiegand, Melanie Siegel, and Josef Ruppenhofer. 2018. Overview of the germeval 2018 shared task on the identification of offensive language, September.

Matthew L. Williams and Pete Burnap. 2015. Cyberhate on social media in the aftermath of Woolwich: A case study in computational criminology and big data. *British Journal of Criminology*, 56(2):211–238.