Data Science Kitchen at GermEval 2021: A Fine Selection of Hand-Picked Features, Delivered Fresh from the Oven

Niclas Hildebrandt, Benedikt Boenninghoff, Dennis Orth, Christopher Schymura
Data Science Kitchen
{firstname.lastname}@data-science-kitchen.de

Abstract

This paper presents the contribution of the Data Science Kitchen at GermEval 2021 shared task on the identification of toxic, engaging, and fact-claiming comments. The task aims at extending the identification of offensive language, by including additional subtasks that identify comments which should be prioritized for fact-checking by moderators and community managers. Our contribution focuses on a feature-engineering approach with a conventional classification backend. We combine semantic and writing style embeddings derived from pre-trained deep neural networks with additional numerical features, specifically designed for this task. Ensembles of Logistic Regression classifiers and Support Vector Machines are used to derive predictions for each subtask via a majority voting scheme. Our best submission achieved macro-averaged F1-scores of 66.8%, 69.9% and 72.5% for the identification of toxic, engaging, and fact-claiming comments.

1 Introduction

In the early years after establishing social media platforms, setting up online discussion forums and installing comment areas on newspapers' websites, a door into this new digital world has been opened, allowing people to interconnect all over the world. Various communication platforms and social networks enabled new ways of sharing information with followers, exchanging opinions between politically interested people, and encouraging debates with the readers. Unfortunately, recent trends revealed the ugly face and adverse effects of these platforms when an increasing number of users make improper, illegal, or abusive use of such digital services (Mathew et al., 2019).

Nowadays, social media platforms are notorious for spreading toxic comments, in which the writers justify violence and discrimination against a person or groups of persons (Munn, 2020). Additionally, a second steadily growing trend is producing and sharing fake news or misinformation, seeking to dominate current discussions, and frame public debates (Mahid et al., 2018).

Both hate speech, fake news and their impact have become very prominent in recent years. However, the tremendous amount of shared and distributed toxic messages on social media platforms make it utterly infeasible to identify and tag or delete poisonous comments manually. The GermEval 2021 shared task tries to encounter this negative trend and motivates participants to work on automated solutions towards safer and more reliable digital rooms of interaction (Risch et al., 2021).

Therefore, the organizers of the task increased the difficulty of the competition by expanding the focus not only on the identification of toxic messages in online discussions but also on distinguishing between engaging and fact-claiming comments. The first task is similar to the GermEval tasks in 2018 (Wiegand et al., 2018) and 2019 (Struß et al., 2019) and deals with identifying toxic comments, including offensive, hateful and vulgar language or ruthless cynism. As novel subtasks, the participants are also invited to identify two additional categories of comments: The second category defines engaging comments, which are annotated as highly relevant contributions by the moderators. The third category concentrates on finding fact-claiming comments that should be considered for a manual fact-check with a higher priority.

2 Task and Data Description

Each subtask of GermEval 2021 is defined as a binary classification problem and all tasks share the same training and test data. The set of training data consists of 3,244 Facebook comments from a German news broadcast page. The anonymized

Proceedings of the GermEval 2021 Shared Task on the Identification of Toxic, Engaging, and Fact-Claiming Comments
co-located with KONVENS

comments were posted in the time span from February to July 2019 and were labeled by trained annotators. Binary labels were provided for each of the three categories. The test data is also extracted from Facebook discussions and include 944 comments. However, these comments had a different discussion topic than the training data. Precision, recall, and macro-averaged F1-score were defined as the relevant evaluation metrics.

2.1 Subtask 1: Toxic Comment Classification

Toxic comments are characterized by their offensive and hateful language, intended to blame other people or groups. For social media and content providers, it is important to detect such comments in a highly automated and scalable way. An example of a toxic comment from the training data of the GermEval shared task is: *"Na, welchem tech riesen hat er seine Eier verkauft..?"*. However, some of the comments which have been labeled as toxic can be quite hard to detect. Examples of such cases are: *"@USER eididei sieh mal an"* or *"ein schöner VW Golf Diesel.."*. Difficulties occur due to irony, subtle overtones and missing contextual information.

2.2 Subtask 2: Engaging Comment Classification

Engaging comments encourage other users to join the discussion, express their opinions and share ideas regarding the topic. They are characterised by being rational, respectful, and reciprocal and hence can foster a constructive and fruitful discussion. The comment *"Wie wär's mit einer Kostenteilung. Schließlich haben beide Parteien (Verkäufer und Käufer) etwas von der Tätigkeit des Maklers. Gilt gleichermassen für Vermietungen. Die Kosten werden so oder soweit verrechnet, eine Kostenreduktion ist somit nicht zu erwarten."* is an example of an engaging comment from the training data.

2.3 Subtask 3: Fact-Claiming Comment Classification

If a platform provider has to prevent the spread of fake news and misinformation, there is the demand of automatically identifying fact-claiming comments to assess their truthfulness. An example of a fact-claiming comment from the training data is the comment *"Dummerweise haben wir in der EU und in der USA einen viel höheren CO2 Fußabdruck als z.B. die Afrikaner oder Inder."*.

3 System Overview

The general system architecture is shown in Fig. 1. As the number of samples in the training dataset provided for GermEval 2021 is rather small, our proposed framework focuses on suitable feature engineering with a conventional classification backend. These features and further implementation details of our system are described in the following.

3.1 Preprocessing

Raw input text is preprocessed in three different processing streams that are handled in parallel. The first stream utilizes the tokenizer of a German BERT model (Chan et al., 2020) and crops the corresponding input text at a maximum length of 512 tokens. The second stream uses the SoMaJo tokenizer (Proisl and Uhrig, 2016) for German language and the third stream passes the raw text to the subsequent feature extraction stage without any preprocessing.

3.2 Feature Extraction

The feature extraction stage focuses on embedding-based features, as well as manually selected, numerical feature representations. Specific feature types are computed using one of the three preprocessing streams described in Sec. 3.1. This specific feature extraction setup was chosen to efficiently combine embedding representations that capture linguistic properties with "hand-crafted" features specifically designed for the GermEval 2021 tasks.

3.2.1 Semantic Embeddings

The first kind of embeddings used in our framework are document embeddings derived from a pre-trained German BERT model (Chan et al., 2020). Specifically, we used the `bert-base-german-cased` implementation from Huggingface[1]. This model was trained on a German Wikipedia dump, the OpenLegalData dump (Ostendorff et al., 2020) and news articles. Average pooling was used to compute 768-dimensional document-level embeddings from the BERT model output.

3.2.2 Writing Style Embeddings

Besides semantic document embeddings, we additionally experimented with neural stylometric embeddings that have been automatically extracted

[1] https://huggingface.co/dbmdz/bert-base-german-cased

Proceedings of the GermEval 2021 Shared Task on the Identification of Toxic, Engaging, and Fact-Claiming Comments
co-located with KONVENS

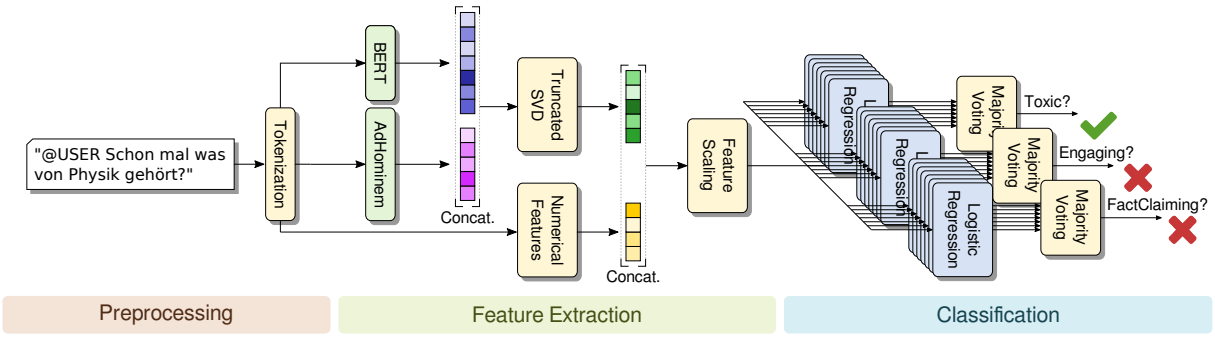


Figure 1: General architecture of the proposed framework to detect toxic, engaging and fact-claiming comments. Yellow boxes denote non-trainable, computational functions and transformations, green boxes represent pretrained models utilized for feature extraction. Trainable models whose parameters are optimized using the challenge dataset are shown in blue.

from the comments. More precisely, we used an extended framework of ADHOMINEM (Boenninghoff et al., 2019) which outperformed all other systems that participated in the PAN 2020 and 2021 authorship verification tasks (Boenninghoff et al., 2021).

The overall framework consists of three components: In a first step, we perform neural feature extraction and deep metric learning (DML) to encode the writing style characteristics of a pair of raw documents into a pair of fixed-length representations, which is realized in the form of a Siamese network. Inspired by (Hochreiter and Schmidhuber, 1997), the Siamese network consists of a hierarchical LSTM-based topology. Next, the obtained representations are fed into a Bayes factor scoring (BFS) layer to compute the posterior probability for this trial. The idea of this second component is to take into account both, the similarity between the questioned documents and the typicality w.r.t. the relevant population represented by the training data. The third component is given by an uncertainty adaptation layer (UAL) aiming to correct possible misclassifications and to return corrected and calibrated posteriors. More details can be found in (Boenninghoff et al., 2021).

To train the model, we prepared a large dataset of Zeit-Online forum comments[2]. Altogether, we collected 9,812,924 comments written by 204,779 authors. Afterwards we split the dataset into training and validation sets. We took 10% of the authors to build the validation set and removed all comments with less than 60 tokens. Due to the fact that the provided dataset of the shared task also contains concise comments, we decided to leave all short comments in the training set. As a result, the datasets are disjoint w.r.t. the authors, i.e., all authors in the validation set have been removed from the training set. During training, we perform data augmentation by resampling new same-author and different-authors pairs in each epoch. Contrary, the pairs of the validation set are sampled once and then kept fixed. Since some authors contribute with hundreds of comments, we limited their influence by sampling not more than 20 comments per author. In summary, the training set contains approximately 234,500 same-author and 244,200 different-authors pairs in each epoch, where, on average, each comment consists of 75.90 ± 68.07 tokens. The validation set contains 15,125 same-author and 18,740 different-authors pairs, where, on average, each comment consists of 126.51 ± 65.31 tokens. Hence, both datasets are nearly balanced.

Table 1: Results for PAN 2021 evaluation metrics.

Model	PAN 2021 Evaluation Metrics					
	AUC	c@1/acc	f_05_u	F1	Brier	Overall
DML	87.4	79.3	81.7	81.0	85.1	82.9
BFS	87.4	79.5	80.5	82.0	85.5	83.0
UAL	87.6	79.5	81.6	81.4	85.6	83.2

We choose the PAN 2021 evaluation metrics to evaluate the performance as described in (Kestemont et al., 2021). Table. 1 summarizes the results, where all three system components are evaluated separately. It can be seen that we achieved overall scores between 82.9 and 83.2 for the components, which is mainly supported by higher values for the AUC and Brier scores. Comparing the c@1, F1 or f_05_u metrics, we generally obtained error rates of approximately 20% on this challenging dataset for a fixed threshold. After training, one part of the neural feature extraction component within the Siamese network is then used to extract the 100-dimensional writing style embeddings for the shared task data.

[2] www.zeit.de

Table 2: Overview of all features utilized in this work that are not based on embeddings.

Feature name	Dim.	Description
NumCharacters	1	Total number of characters, including white spaces.
NumTokens	1	Total number of tokens, after splitting at white spaces.
AverageTokenLength	1	Average number of characters in all tokens.
TokenLengthStd	1	Standard deviation of the number of characters in all tokens.
StopwordRatio	1	Number of stop words divided by the number of tokens.
ExclamationMarkRatio	1	Number of exclamation marks divided by the number of characters.
NumReferences	1	Number of hyperlinks in the comment.
NumMediumAdressed	1	Number of @MEDIUM mentions in the comment.
NumUserAdressed	1	Number of @USER mentions in the comment.
AverageEmojiRepetition	1	Average repetition number of emojis used in the comment.
SpellingMistakes	17	Number of specific grammar and spelling mistakes, cf. Sec. 3.2.3.
SentimentBERT	3	Sentiment scores of a pre-trained BERT model (Guhr et al., 2020).

3.2.3 Additional Numerical Features

In addition to the semantic and writing style embeddings, we integrated a set of specifically designed numerical features into our framework. An overview of these features, their dimensionality and corresponding descriptions is given in Tab. 2. We applied the natural logarithm to all strictly-positive numerical features.

The first group of features, NumCharacters, NumTokens, AverageTokenLength and TokenLengthStd, were chosen to reflect general structural properties of the comments in the dataset. In addition, we use the StopwordRatio and ExclamationMarkRatio features to explicitly reflect task-related semantic properties in the dataset. These task-specific features are accompanied by additional count-based features NumMediumAdressed, NumUserAddressed, NumReferences and AverageEmojiRepetition. We also included the scores (corresponding to the classes "positive", "neutral" and "negative") of a BERT model for sentiment classification trained on 1,834 million German-language samples derived from various sources (Guhr et al., 2020) as a dedicated SentimentBERT feature.

Lastly, we included an 17-dimensional feature denoted as SpellingMistakes into our set of additional features. This feature represents spelling and grammar mistakes from 17 different categories. We used a Python wrapper from the open-source grammar checker LanguageTool[3] to derive this feature. In particular, the following classes of mistakes were considered: *Typography*, *punctuation*, *grammar*, *upper/lowercase*, *support in punctuation*, *colloquialism*, *compounding*, *confused words*, *redundancy*, *typos*, *style*, *proper nouns*, *idioms*, *recommended spelling*, *miscellaneous*, *double punctuation*, *double exclamation mark*. For every category, we counted the number of mistakes and divided them by the number of tokens in the respective comment.

[3] https://languagetool.org/

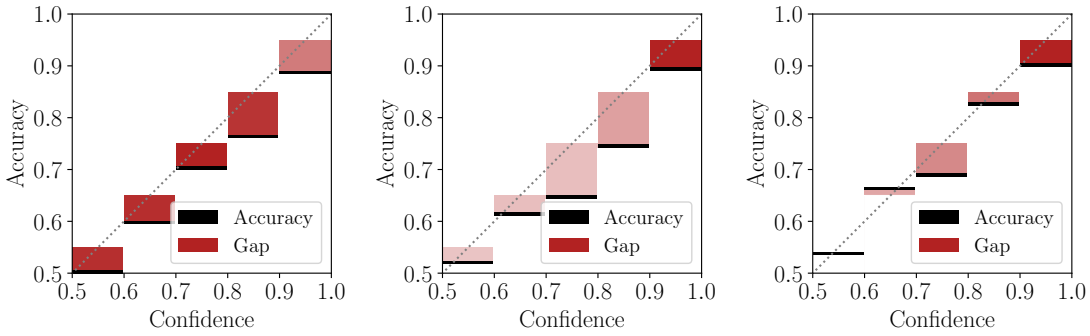
3.3 Classification Pipeline

The classification pipeline used in this work is depicted in Fig. 1. The semantic and writing style embedding features described in Secs. 3.2.1 and 3.2.2 are concatenated, yielding a 868-dimensional joint embedding vector. A truncated singular-value decomposition (SVD) (Halko et al., 2011) is applied on this vector to reduce its dimensionality for subsequent processing. The number of dimensions kept is treated as a hyperparamter during training, cf. Sec. 4. The reduced joint embedding vector is then concatenated with the 28-dimensional vector of additional numerical features. The resulting vector is standardized to zero-mean and unit variance and serves as input to the classification stage.

We use Logistic Regression (Berkson, 1944; Haggstrom, 1983) and Support Vector Machines (SVMs) (Boser et al., 1992) with radial basis function (RBF) kernel as base classifiers within individual ensembles. One ensemble of binary classifiers is utilized for each subtask. Each classifier in the ensembles is trained using a subset of the provided training data via a cross-validation setup, cf. Sec. 4. A hard majority-voting scheme is used in each ensemble to obtain the predicted labels.

4 Evaluation

Our framework is trained using a specific cross-validation and hyperparameter tuning scheme, which is described in the following.



(a) Subtask 1 (b) Subtask 2 (c) Subtask 3

Figure 2: Reliability diagrams of our first submission for all three subtasks (see Section 4.1). The red bars are drawn darker for bins with a higher number of samples.

4.1 Evaluation Metrics

Precision, recall and macro average F1-score are used for model evaluation (Opitz and Burst, 2021) since they represents the evaluation metrics of the GermEval 2021 shared task. Additionally, we assess the calibration properties of our model by determining the expected calibration error (ECE) as well as the maximum calibration error (MCE), where the confidence interval is discretized into a fixed number of M bins (Naeini et al., 2015). The ECE is then computed as the weighted macro-averaged absolute error between confidence and accuracy of all bins,

$$\mathrm{ECE} = \sum_{m=1}^{M} \frac{|B_m|}{N} \left|\mathrm{acc}(B_m) - \mathrm{conf}(B_m)\right|, \quad (1)$$

where N is the total number of samples and $\mathrm{acc}(B_m) - \mathrm{conf}(B_m)$ is the difference between the actual accuracy and classifier confidence within a fixed-size bin B_m in the confidence interval. Note that all confidence values lie within the interval $[0.5, 1]$,since we are dealing with binary classification tasks. Hence, to obtain confidence scores, the output predictions p are transformed w.r.t. to the estimated subtask label, showing $\mathrm{conf} = p$ if the $\mathrm{acc} >= 0.5$ and $\mathrm{conf} = 1 - p$ if $\mathrm{acc} < 0.5$. The MCE returns the maximum absolute error, given as

$$\mathrm{MCE} = \max_{m \in 1,\ldots,M} \left|\mathrm{acc}(B_m) - \mathrm{conf}(B_m)\right|. \quad (2)$$

We further display the reliability diagrams in Fig. 2 which will be discussed in Section 5.

4.2 Experimental Setup

Our experimental setup involves dedicated model selection and hyperparameter tuning. The training set performance is evaluated in a stratified K-Fold cross validation setup preserving the class label distribution among all folds. One of the K folds is used as validation set. We utilized a 7-fold cross-validation scheme and computed the evaluation metrics described in Sec. 4.1 on the validation set of each fold.

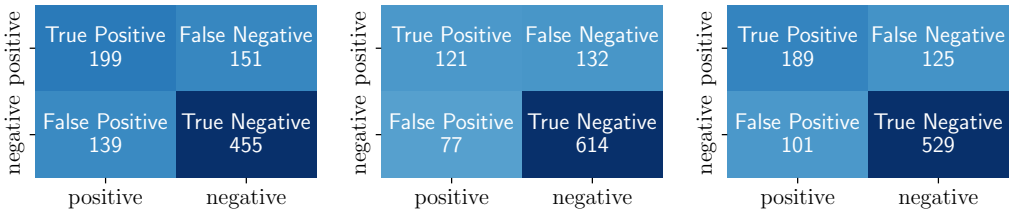
For submission one and two there are 7 logistic regression models for each subtask trained on different folds and stacked together in a voting ensemble returning the prediction of the majority. On each fold the L2-regularisation strength `C` and the number of features coming from the SVD dimension reduction are tuned with respect to the macro averaged F1 score over all subtasks. This means that hyperparameters may be slightly different from fold to fold but all three models trained on the same fold get the same hyperparameters – regardless the classification task.

Submission three uses a similar approach but the logistic regression models are replaced by SVMs having the same fold-wise hyperparameter tuning as mentioned above. In addition, task-wise tuned SVMs are added to the ensemble. Doubling the number of models and including a higher level of customisation to the task. The task-wise tuning includes optimisation of `kernel`, L2 regularisation strength `C`, `class weight`(whether or not to weight `C` with the class label distribution) and the kernel coefficient `gamma` as defined in the sklearn library (Pedregosa et al., 2011).

Hyperparameter tuning is performed with Bayesian optimisation using the Optuna library (Akiba et al., 2019). The macro average F1-score is chosen as optimisation target and the best hyperparameters among 100 trails are used in the ensemble.

Table 3: Final submission results on the test set including the calibration metrics for the first submission.

Run	Subtask 1					Subtask 2					Subtask 3				
	P	R	F1	ECE	MCE	P	R	F1	ECE	MCE	P	R	F1	ECE	MCE
Submission 1	65.95	63.67	64.79	5.5	8.0	69.70	67.78	68.72	6.9	10.4	73.25	71.44	72.34	3.5	6.0
Submission 2	64.89	62.71	63.78	–	–	69.26	67.43	68.33	–	–	73.39	71.52	72.44	–	–
Submission 3	66.98	66.73	66.85	–	–	71.71	68.34	69.98	–	–	73.03	72.08	72.55	–	–



(a) Subtask 1 (b) Subtask 2 (c) Subtask 3

Figure 3: Confusion matrices for submission 3.

5 Results and Discussion

The final submission results are provided in Table 3. Unexpectedly, the identification of toxic comments turns out to be the most challenging subtask while the detection of fact-claiming comments achieved the highest F1-score. This confirms our observations during hyperparameter tuning. For instance, the F1-score for the third submission after cross validation are given by 66.31 ± 1.76, 75.12 ± 2.07 and 74.68 ± 2.67 for subtasks 1-3, respectively. A comparison of our cross validation performance with the results on the test set shows two interesting findings: On the one side, we obtained very robust results of subtasks 1 and 3. On the other side, subtask 2 struggles with over-fitting effects.

In addition, Fig. 3 displays the confusion matrices of our third submission (representative for all submissions). It can be seen for all subtasks that the ratio of wrongly classified positively labeled samples is significantly larger than for negatively labeled samples. This behavior is supported by the reliabilty diagrams[4] in Fig. 2, where our submission delivers *over-confident* scores (i.e. conf $>$ acc) in nearly all bins. As a results, the higher proportion of wrongly classified comments for positively labeled comments leads to a lower performance in terms of the F1-score.

Finally, we visualize an estimated probability density function of the first submission using a non-parametric Gaussian kernel density estimator[5] in Fig. 4. Ideally, we would expect a bimodal probability density function. However, the plot shows that the system clearly tends towards self-confident predictions close to zero. But in regions closer to one, the systems behave more hesitant. This effect can be explained by the imbalanced distribution of the class labels.

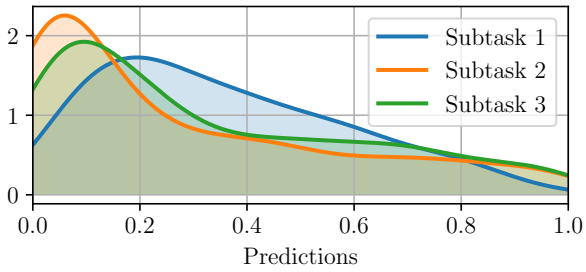


Figure 4: Gaussian kernel density estimates for the distributions of the first submission (bandwidth= 0.08).

6 Conclusions

Within this contribution to the shared task of the GermEval 2021 we have developed a modular feature extraction scheme which incorporates semantic and writing style embeddings as well as task specific numerical features. Less complex algorithms like logistic regression models and SVMs converge converge faster than complex models like deep neural networks and therefore need less training data. The combination with automated hyperparameter tuning and dimension reduction as well as the final agglomeration of multiple models in voting ensembles allow to achieve an macro-averaged F1-scores of 66.8%, 69.9% and 72.5% for the identification of toxic, engaging, and fact-claiming comments.

[4] https://github.com/hollance/reliability-diagrams

[5] https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KernelDensity.html

References

Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. 2019. Optuna: A next-generation hyperparameter optimization framework.

Joseph Berkson. 1944. Application of the logistic function to bio-assay. *Journal of the American Statistical Association*, 39(227):357–365.

B. Boenninghoff, S. Hessler, D. Kolossa, and R. M. Nickel. 2019. Explainable Authorship Verification in Social Media via Attention-based Similarity Learning. In *IEEE International Conference on Big Data*, pages 36–45.

Benedikt Boenninghoff, Dorothea Kolossa, and Robert M. Nickel. 2021. Self-Calibrating Neural-Probabilistic Model for Authorship Verification Under Covariate Shift. In *12th International Conference of the CLEF Association (CLEF 2021)*. Springer.

Bernhard E. Boser, Isabelle M. Guyon, and Vladimir N. Vapnik. 1992. A training algorithm for optimal margin classifiers. In *Proceedings of the Fifth Annual Workshop on Computational Learning Theory*, COLT '92, page 144–152, New York, NY, USA. Association for Computing Machinery.

Branden Chan, Stefan Schweter, and Timo Möller. 2020. German's next language model. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6788–6796, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Oliver Guhr, Anne-Kathrin Schumann, Frank Bahrmann, and Hans Joachim Böhme. 2020. Training a broad-coverage german sentiment classification model for dialog systems. In *Proceedings of The 12th Language Resources and Evaluation Conference*, pages 1620–1625, Marseille, France. European Language Resources Association.

Gus W. Haggstrom. 1983. Logistic regression and discriminant analysis by ordinary least squares. *Journal of Business & Economic Statistics*, 1(3):229–238.

N. Halko, P. G. Martinsson, and J. A. Tropp. 2011. Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions. *SIAM Review*, 53(2):217–288.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Computation*, 9(8):1735–1780.

Mike Kestemont, Efstathios Stamatatos, Enrique Manjavacas, Janek Bevendorff, Martin Potthast, and Benno Stein. 2021. Overview of the Authorship Verification Task at PAN 2021. In *CLEF 2021 Labs and Workshops, Notebook Papers*. CEUR-WS.org.

Zaitul Iradah Mahid, Selvakumar Manickam, and Shankar Karuppayah. 2018. Fake news on social media: Brief review on detection techniques. In *2018 Fourth International Conference on Advances in Computing, Communication Automation (ICACCA)*, pages 1–5.

Binny Mathew, Ritam Dutt, Pawan Goyal, and Animesh Mukherjee. 2019. Spread of hate speech in online social media. WebSci '19, page 173–182, New York, NY, USA. Association for Computing Machinery.

Luke Munn. 2020. Angry by design: toxic communication and technical architectures. *Humanities and Social Sciences Communications*, 7(53):229–238.

Mahdi Pakdaman Naeini, Gregory Cooper, and Milos Hauskrecht. 2015. Obtaining well calibrated probabilities using bayesian binning.

Juri Opitz and Sebastian Burst. 2021. Macro f1 and macro f1.

Malte Ostendorff, Till Blume, and Saskia Ostendorff. 2020. Towards an open platform for legal information. In *Proceedings of the ACM/IEEE Joint Conference on Digital Libraries in 2020*, JCDL '20, page 385–388, New York, NY, USA. Association for Computing Machinery.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Thomas Proisl and Peter Uhrig. 2016. SoMaJo: State-of-the-art tokenization for German web and social media texts. In *Proceedings of the 10th Web as Corpus Workshop (WAC-X) and the EmpiriST Shared Task*, pages 57–62, Berlin. Association for Computational Linguistics (ACL).

Julian Risch, Anke Stoll, Lena Wilms, and Michael Wiegand. 2021. Overview of the GermEval 2021 shared task on the identification of toxic, engaging, and fact-claiming comments. In *Proceedings of the GermEval 2021 Shared Task on the Identification of Toxic, Engaging, and Fact-Claiming Comments co-located with KONVENS*, pages 1–12.

Julia Maria Struß, Melanie Siegel, Josep Ruppenhofer, Michael Wiegand, and Manfred Klenner. 2019. Overview of germeval task 2, 2019 shared task on the identification of offensive language. In *Proceedings of the 15th Conference on Natural Language Processing (KONVENS 2019)*, pages 354–365, Erlangen, Germany. German Society for Computational Linguistics & Language Technology.

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