Effective Detection of Hate Speech Spreaders on Twitter

Notebook for PAN at CLEF 2021

Julian Höllig[a], Yeong Su Lee[a], Nina Seemann[a] and Michaela Geierhos[a]

[a]*Research Institute CODE, Bundeswehr University Munich, Neubiberg, Germany*

Abstract

In this paper, we summarize our participation in the task of "Profiling Hate Speech Spreaders on Twitter" at the PAN@CLEF Conference 2021. Our models obtained an average accuracy of 76% (79% for Spanish and 73% for English). For English, we used a Linear Support Vector Machine with tf-idf features on noun chunk level, while for Spanish we used a Ridge Classifier with simple counts on noun chunk level. Both classifiers were fed with additional features obtained from a Convolutional Neural Network.

Keywords

author profiling, hate speech, noun chunks

1. Introduction

In recent years, there has been growing awareness that hate speech became an increasing issue in social media, which offers anonymity and virality to authors of hateful posts. On average, Twitter had 199 million daily active users in the first quarter of 2021, compared to 166 million active users counted the year before, which is an increase of almost 20 percent [1]. These developments caused political forces and social media providers to act. For example, the EU initiated measures such as the European Council's "No Hate Speech Movement", which aims at mobilizing online consumers to take action against hate [2]. The EU, with the involvement of YouTube, Twitter, Facebook, and Microsoft, has also drafted a "Code of conduct on countering illegal hate speech online" [3], in which these companies commit to check hate speech notifications within 24 hours [4]. However, the most effective and efficient way to combat hate speech is through its automatic detection, where machine learning applications can play a crucial role.

In the PAN shared task [5], international researchers focus on modeling such applications to identify hate speech spreaders on Twitter. Compared to other hate speech detection tasks [6, 7], the data for this task consist of tweet collections belonging to the same author, each representing a sample in the dataset. One profound challenge in detecting hate speech is its unclear definition. Table 1 compares the attitudes of political, social media, and scientific representatives on four important questions about the definition of hate speech [8]. The contents of the table were retrieved from official sources of the institutions and from scientific papers [8]. While there

CLEF 2021 – Conference and Labs of the Evaluation Forum, September 21–24, 2021, Bucharest, Romania

julian.hoellig@unibw.de (J. Höllig); yeongsu.lee@unibw.de (Y. S. Lee); nina.seemann@unibw.de (N. Seemann); michaela.geierhos@unibw.de (M. Geierhos)

Table 1
Attitudes of political, social media, and scientific representatives on hate speech. [8]

Source	Hate speech is to incite violence or hate	Hate speech is to attack or disparage	Hate speech has specific targets	Humor has a specific status
EU code of conduct	Yes	No	Yes	No
ILGA	Yes	No	Yes	No
Scientific paper	No	Yes	Yes	No
Facebook	No	Yes	Yes	Yes
YouTube	Yes	No	Yes	No
Twitter	Yes	Yes	Yes	No

is agreement that hate speech is directed to specific targets, the attitudes differ regarding the influence of humor, whether or not hate speech is intended to incite hate, and whether or not hate speech is intended to attack and disparage. This leads to critically different definitions of hate speech and how to combat it. For example, according to Table 1, YouTube would not define a verbal attack on a person as hate speech, while inciting violence against the same person would be considered as hate speech. Facebook would take the opposite position, according to Table 1. Due to the broad definition of hate speech, it is difficult to find large harmonized data collections, since annotators often have low agreement when building new collections [9, 10, as cited in [8]]. Consequently, it is challenging to establish a common standard for modeling hate speech so far.

The paper is organized as follows. In Section 2, we present relevant related work. In Section 3, we explain our methods in detail and present the evaluation of the final models in Section 4. Finally, we conclude in Section 5.

2. Related Work

In recent years, the detection of hate speech has been of great interest for many researches. As a result, there is a vast literature on this topic. In the following, we will only focus on some other shared tasks and their results.

Mandl et al. [11] describe the identification of hate speech and offensive content in Indo-European languages[1] at FIRE 2019. There were three subtasks: subtask A was the coarse-grained binary classification into non Hate-Offensive and Hate & Offensive (HOF). If a post was classified as HOF, then it was handled by subtask B, which further classified it into either hate speech, offensive, or profane. Subtask C addressed the targeting and non-targeting of individuals, groups, or others when a post was classified as HOF. For example, the team with the best performance on the English data achieved a macro F1 score of 78.82% and a weighted F1 score of 83.95% for subtask A, a macro F1 of 54.46% and a weighted F1 of 72.77% for subtask B, and a macro F1 of 51.11% and a weighted F1 of 75.63% for subtask C.

HatEval [12] consists of detecting hateful content in Twitter posts for English and Spanish. There were two subtasks: subtask A focused on detecting hate speech against immigrants and

[1]The three languages provided were English, Hindi, and German.

women, i.e., a binary classification into hateful or not. In subtask B, a fine-grained classification had to be performed. Hateful tweets had to be further classified in terms of (i) aggressive attitude, i.e., is a tweet aggressive or non-aggressive, and (ii) target classification, i.e., is a specific target harassed or a generic group. Both tasks in subtask B are binary. For subtask A, the best systems obtained a macro-averaged F1 score of 0.651 for English and a macro-averaged F1 score of 0.73 for Spanish. For subtask B, the best systems achieved an Exact Match Ratio (EMR) of 0.570 for English and 0.705 for Spanish.

Bosco et al. [6] describe the hate speech shared task at the Sixth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian (Evalita) in 2018. Both tasks, Evalita and PAN, focused on detecting hate speech on social media (Facebook and Twitter). However, Evalita targeted hate speech detection at the post level, i.e., tweets. The PAN task focused on identifying an author as hate speech spreader based on a collection of his/her tweets, which introduced additional fuzziness to the challenging task of defining and detecting hate speech. Nine out of ten Evalita participants used external resources to improve their systems, such as pre-trained embeddings, dictionaries, and datasets related to the task. The winning team achieved an F1 score of almost 80% by using additional data from a subjectivity and polarity lexicon and the SENTIPOLC dataset on sentiment analysis along with linear SVM and BiLSTM models. In our work, we also successfully experimented with additional external data. However, the data was not directly used to identify hate speech spreaders, but was used to score the tweets themselves to create a 'hate weight' for each author.

Struß et al. [13] summarize the 2019 GermEval shared task on identifying offensive language in Twitter data, where 'offensive' is defined as insulting, abusive, or profane language. The shared task was divided in three subtasks: (1) a binary classification into offensive and non-offensive tweets, (2) a multi-classification into insulting, abusive, profane, and non-offensive tweets, and (3) a binary classification of offensive tweets into implicitly and explicitly offensive. The dataset consisted of 7,000 tweets in total (4,000 train set, 3,000 test set). For subtask (2), the offensive tweets were divided into the three indicated groups. For subtask (3), 2,900 offensive tweets were split into 400 implicitly and 2,500 explicitly offensive tweets. The best performing system on all subtasks was a BERT model, which achieved 77%, 54%, and 73% in macro average F1 score (on subtasks (1), (2), and (3)). It was pre-trained on six million German tweets and fine-tuned on the GermEval data. The average performances of all participants obtained on the subtasks were 72%, 47%, and 67%.

3. Methods

We experimented with different approaches and methods, using both deep learning, i.e. neural networks, and machine learning, i.e. more traditional algorithms. In the following sections, we provide an overview of the PAN dataset and its challenges before describing in detail the steps taken to obtain our final results for the task.

Table 2
Overview of the dataset size.

Language	Authors		Tweets
	# Hate Spreaders	# Legitimate Users	
en	100	100	40,000
es	100	100	40,000

3.1. Dataset

For both English and Spanish, the organizers provided us with a dataset containing 200 tweets for 200 authors each, indicating whether or not the author is considered as hate speech spreader. This classifies 100 authors as hate speech spreaders and 100 as legitimate users. In total, the dataset contains 40,000 tweets for each language. An overview of the dataset can be found in Table 2. What makes this author profiling task so challenging is the fact that the tweets collected for an author are neither all hate speech nor all harmless. Hence, not all of the 200 tweets per hate speech spreader are hateful per se. Information on the annotation scheme or threshold (i.e., the minimum number of hateful tweets per author) for classifying an author as hate speech spreader was not provided at the time of the competition.

3.2. Additional Features

Inspired by the challenging mix of hate speech and harmless tweets per author, we developed additional features to weigh the amount of hateful content produced by each author. Therefore, we relied on external data, which we describe in Section 3.2.1. In recent years, many NLP applications — including text classification tasks — have been significantly improved by the use of deep learning algorithms. Hence, we decided to train a Convolutional Neural Network (CNN, [14]) with external data and applied the resulting model on the PAN data at the tweet level to generate additional features. In the following subsections, we describe this process in more detail.

3.2.1. External data

To create the additional features described in Section 3.2, we searched for external data containing hate speech or hate speech related concepts such as offensive language. Since the PAN dataset consists of tweets, we looked for external data also retrieved from Twitter. Our search resulted in several good sources for English. We chose the data from (i) CONAN [15], (ii) Davidson et al. [16], (iii) HASOC track [11], and (iv) SemEval-2019 Task 5 HatEval [12]. Unfortunately, there was not much data available for Spanish, so we only used the Spanish part of the HatEval dataset [12]. An overview of the sizes and the percentage of offensive tweets in the datasets is given in Table 3.

Table 3
Overview of the external data used for training the CNN.

Lang	dataset	Size	% Offensive
en	CONAN	1,288	1.00
	Davidson	24,802	0.06
	HASOC	7,005	0.36
	hatEval	13,000	0.40
es	hatEval	6,600	0.40

3.2.2. Training of the CNN

We used the Tensorflow/Keras API[2] for the implementation of the CNN. Since the input for neural networks does not require much preprocessing, we simply lowercased the external dataset and removed unwanted characters, symbols, and emojis. We implemented a character-level model with 49 features for English and 57 features for Spanish. The number of features is determined by the number of different characters present after preprocessing. After examining the character length of all tweets, we set the maximum sequence length to 300 per tweet. Keras provides a tokenizer that converts the input into a list of integers (similar to the bag-of-words approach) and a method to convert these integers into a fixed-length vector. This means that tweets containing more than 300 characters were reduced in size and shorter tweets were padded with zeros to maximum length. We trained the CNN with the following setting:

- filter size: [5,6,7]
- number of filters: 100
- activation: ReLU
- output: sigmoid

We used the Adam optimizer [17] and trained the CNN for 80 epochs. After the last epoch, the model had an accuracy of 91.01% on the English external data and 92.12% on the Spanish external data.

3.2.3. Applying the model to the PAN data

For each author, we let the model obtained by the CNN predict the class for each of his/her tweets. We recorded the values of the predictions as 'HateCounts' and 'LoveCounts'. After classifying all tweets, we used the majority vote on these counts to predict whether an author was a hate speech spreader or not. In numbers, whenever 'HateCount' was > 100, the author was classified as hate speech spreader and vice versa. Unfortunately, this resulted in an accuracy of about 0.5 for both languages. So we decided to iteratively lower the threshold of 'HateCount' from 100 to 0 to get better accuracy. For English, a threshold of 48 gave the best accuracy of 67.58%, while we obtained the best accuracy of 71.5% with a threshold of 33 for Spanish. Since these performances were not convincing, we decided to move to more traditional machine learning algorithms (see

[2] https://www.tensorflow.org/api_docs/python/tf/keras

Section 3.3). Unlike deep learning models, traditional machine learning models have no sequence length limit, so we could use them to process all tweets per author at once. However, since the 'HateCounts'/'LoveCounts' showed at least some influence on the classification of hate speech spreaders, we kept them as additional features for the subsequent experiments with the traditional models. Furthermore, we obtained the class probability calculated by the CNN for each tweet and also stored the mean probabilities for each author ('ProbMean').

3.3. Experiments

In the following sections, we describe our experiments using traditional methods that led us to our final models. All experiments were implemented in Python using the scikit-learn library[3].

3.3.1. Experimental setup A

In our first setup, we tested five algorithms for different features. As preprocessing, we combined all tweets of an author into one document. To mark the beginning of a new tweet, we added a special start-of-tweet token. Furthermore, the data was lowercased. As features, we used simple counts of bag-of-words and tf-idf at the word and character level. To reduce the feature space, we limited the maximum number of features to 5,000. Additional features were not considered for these experiments. We split the data into 70% for training and 30% for testing, using three different random states for splitting. The experimental results are presented in Table 4. The values shown are the mean accuracy for the different data splits. The results were not sufficient, but gave us a reasonable basis on which to improve further.

Table 4
Mean accuracy of three runs with different random states on the English dataset.

Algorithm	Count	tf-idf		
		word		character
		1-gram	2- and 3-gram	2- and 3-gram
Multinomial Naive Bayes	59.67	53.00	56.33	58.33
Logistic Regression	68.67	56.00	52.67	53.67
Support Vector Machine	51.67	59.00	53.30	54.30
Random Forest	57.00	61.67	57.00	60.00
Extreme Gradient Boosting	63.00	63.67	53.00	59.33

3.3.2. Experimental setup B

For our second setup, we used the same setup as for experimental setup A, but fed the 'LoveCounts' and the 'ProbMeans' into the models as additional features to improve our performance. Since the PAN dataset contains only 200 instances in total, we also applied oversampling to the training

[3] https://scikit-learn.org/stable/index.html

data using the BorderlineSMOTE library from imbalanced-learn[4]. In doing so, we performed oversampling for both classes rather than just one class (as usually done by SMOTE). Preliminary results showed that an oversampling of 500 gave the best results. Again, we ran experiments with the same five algorithms, but this time with five different random states for splitting into training and test sets. For English, the results obtained are shown in Table 5. The results show that both the additional features and oversampling increased the model performances.

Table 5
Mean accuracy of five runs with different random states and SMOTE=500. LC is the count of non-hate tweets and PM is the mean of the probabilities obtained from the CNN on the English dataset.

Algorithm	Count	tf-idf		
		word		character
	LC + PM	1-gram LC + PM	2- and 3-gram LC + PM	2- and 3-gram LC + PM
Multinomial Naive Bayes	65.0	66.2	51.8	52.6
Logistic Regression	62.0	65.6	63.0	63.4
Support Vector Machine	61.8	68.2	64.4	62.8
Random Forest	63.0	57.4	63.0	60.8
Extreme Gradient Boosting	62.8	59.8	58.8	61.8

3.3.3. Experimental setup C

For our third setup and final experiments, we optimized our data preprocessing to achieve further improvements. The following experiments were performed for both English and Spanish, but for brevity we present only the English results. To this end, we first removed stop words by using the stop word list provided by the spaCy library[5]. Second, we used an emoji sentiment recognizer[6] based on the research by Novak et al. [18]. We replaced positive emojis with 'EMOJI-positive', negative emojis with 'EMOJI-negative', and the remaining ones with 'EMOJI'[7]. Finally, we extracted all noun chunks from the tweets by again using the spaCy library[8]. These noun chunks were then used as input for our experiments. Each word of a noun chunk was lemmatized.

As shown in Table 6, we added more algorithms in addition to those from the previous setups. We used count and tf-idf vectors based on noun chunks as features, both in combination with the 'LoveCount' and 'ProbMean' features, as these were the most promising from our previous results. We did not limit the number of features for this setup. The mean accuracy for the English test set can be found in Table 6.

[4]https://imbalanced-learn.org/stable/

[5]https://spacy.io/usage/spacy-101#language-data

[6]https://github.com/FLAIST/emosent-py. This was adapted, further developed, and made available for our work by D. Schwimmbeck from the Research Institute CODE, Bundeswehr University Munich, Germany.

[7]We did not distinguish between one or more consecutive emojis: All consecutive occurrences of the same emoji were replaced by a corresponding counterpart.

[8]https://spacy.io/usage/linguistic-features#noun-chunks

Table 6
Mean accuracy of five runs with different random states on the English dataset.

Algorithm	Count + LC + PM	tf-idf + LC + PM
Multinomial Naive Bayes	60.00	60.33
Logistic Regression	62.33	68.66
Linear Support Vector Machine	60.67	71.00
Random Forest	61.33	54.67
Extreme Gradient Boosting	57.67	58.00
Decision Tree Classifier	53.67	56.67
Bernoulli Naive Bayes	57.33	57.33
Complement Naive Bayes	60.00	62.33
Stochastic Gradient Descent Classifier	63.33	66.33
Passive Agressive Classifier	62.00	68.00
Ridge Classifier	62.00	71.33
Perceptron	61.00	66.33
KNeighbors Classifier	59.33	59.33
Nearest Centroid	59.00	65.00
Support Vector Machine	58.67	67.00

3.3.4. Beyond n-gram features

As shown in Table 6, most models with tf-idf on noun chunks outperform their counterparts with bag-of-words count. Moreover, the models of the Ridge Classifier and the Linear SVM show remarkable improvements. Hence, we compared the feature importance for tf-idf vectors from n-gram and noun chunks models. Table 7 shows the five most positive and negative features with their weights[9]. While all n-gram models indicate the importance of the words *president* and *trump*, they are not listed in the five most important features of the noun chunks model[10]. On the other hand, it should be also noted that the word *white* alone receives a high weighting in the 1-gram and 1- & 2-gram models; the word sequence *white people* makes more sense in the 2- & 3-gram and noun chunks models.

Table 7
Feature comparison of n-grams and noun chunks.

1-gram	1- & 2-gram	2- & 3-gram	noun chunks
0.8890 president	0.6595 president	0.3317 president trump	0.7242 america
0.8189 white	0.5830 white	0.3019 white people	0.5982 people
0.7803 people	0.5420 trump	0.2826 vote supports	0.5747 white people
0.7361 border	0.4968 people	0.2506 laughing ent	0.5553 em
0.7026 niggas	0.4860 border	0.2506 emoji laughing ent	0.5405 illegal
-0.5680 automatically	-0.4994 automatically checked	-0.5223 automatically checked	-1.0312 person
-0.5355 bbc	-0.4882 bbc	-0.5013 emoji positive	-0.7812 y'
-0.5326 amp	-0.4020 cum	-0.4239 emoji negative	-0.6077 work
-0.5247 yal	-0.3851 sagittarius	-0.3539 followed automatically	-0.4635 bbc
-0.4966 cum	-0.3831 automatically	-0.3539 followed automatically checked	-0.4578 sagittarius

[9] n-grams were obtained by the standard word analyzer with stop words elimination for English, while the noun chunks by spaCy noun chunks. The first word was removed from them if it is a stop word.

[10] The word sequence *president trump* occurs only in the 18th position.

3.3.5. spaCy noun chunks and feature extraction

Table 7 shows some awkward noun chunks like *illegal* and *y'*. Therefore, we examined the posts containing these words in more detail. The word *illegal* occurs in 185 different posts. However, among these, the word *illegals* occurs in 54 different posts in 21 training documents. The noun chunk *illegal* refers to this use of the word. Of these, only 4 documents were classified as non-hate speech spreader. The other 21 documents were classified as hate speech spreader. From this fact, it can be concluded that the word *illegal* could be weighted highly for the task. In the following, some examples from the training documents are given:

- *The illegals took most of the blacks jobs!*
- *This is why I don't want more illegals in the USA until we take care of ALL our Vets!*
- *Why are you protecting illegals?*

For example, the spaCy noun chunk module tagged *The illegals* as noun chunks from the first sentence. But as described in 3.3.4, the first word *The* is a stop word and was removed (see 3.3.3), and the word *illegals* has been lemmatized to *illegal*[11].

The word *y'* occurs only in the combination of *y'all* in 266 different posts, and the spaCy noun chunk module separates the word *y'* as noun chunk from the word *all*. For example, spaCy assigns noun chunks to three units (*I*, *y'*, *college*) from the following sentence taken from a training post: *I can see why some of y'all ain't go to college*.

4. Evaluation on the PAN Test Sets

For the final evaluation on the test set provided by the PAN2021 shared task organizers, we applied the following algorithms and settings:

- English: Linear SVM with tf-idf vectors on noun chunks
- Spanish: Ridge Classifier with count vectors on noun chunks

On the English test set, we achieved an accuracy of 73%, on the Spanish test set an accuracy of 79%. The overall average accuracy for both languages was 76%. According to the PAN2021 Overview [19], this corresponds to rank 8. Due to technical issues, we could not use TIRA[20]. We sent our results to the organizers by email.

5. Conclusion

By retrieving noun chunks from the data, we employed a specific linguistic feature that proved its suitability for the classification task. As shown, it outperforms most methods based on n-gram features. In contrast to n-gram features, noun chunks form linguistically meaningful units and are therefore more comprehensible. Furthermore, we have developed the additional features 'LoveCount' and 'ProbMean', which add a kind of a 'hate weight' to each author.

In future work, we want to explore how linguistic units such as phrasal expressions, and specifically predicate-argument structures, can be obtained and embedded, and to what extent they can contribute to the classification and other tasks related to textual content.

[11] https://spacy.io/usage/linguistic-features#lemmatization

Acknowledgments

This research is partially funded by dtec.bw – Digitalization and Technology Research Center of the Bundeswehr within the project MuQuaNet.

References

[1] Yahoo! Finance, Twitter Announces First Quarter 2021 Results, 2021. URL: https://s22.q4cdn.com/826641620/files/doc_financials/2021/q1/Q1'21-Earnings-Release.pdf.

[2] Council of Europe, No Hate Speech Youth Campaign, 2017. URL: https://www.coe.int/en/web/no-hate-campaign.

[3] European Commission, The EU Code of conduct on countering illegal hate speech online, 2020. URL: https://ec.europa.eu/info/policies/justice-and-fundamental-rights/combatting-discrimination/racism-and-xenophobia/eu-code-conduct-countering-illegal-hate-speech-online_en.

[4] A. Hern, Facebook, YouTube, Twitter and Microsoft sign EU hate speech code, 2016. URL: https://www.theguardian.com/technology/2016/may/31/facebook-youtube-twitter-microsoft-eu-hate-speech-code.

[5] F. Rangel, G. L. D. L. P. Sarracén, B. Chulvi, E. Fersini, P. Rosso, Profiling Hate Speech Spreaders on Twitter Task at PAN 2021, in: CLEF 2021 Labs and Workshops, Notebook Papers, CEUR-WS.org, 2021.

[6] C. Bosco, F. Dell'Orletta, F. Poletto, M. Sanguinetti, M. Tesconi, Overview of the EVALITA 2018 hate speech detection task, in: T. Caselli, N. Novielli, V. Patti, P. Rosso (Eds.), Proceedings of the Sixth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2018) co-located with the Fifth Italian Conference on Computational Linguistics (CLiC-it 2018), Turin, Italy, December 12-13, 2018, volume 2263 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2018. URL: http://ceur-ws.org/Vol-2263/paper010.pdf.

[7] H. Al-Khalifa, W. Magdy, K. Darwish, T. Elsayed, H. Mubarak (Eds.), Proceedings of the 4th Workshop on Open-Source Arabic Corpora and Processing Tools, with a Shared Task on Offensive Language Detection, European Language Resource Association, Marseille, France, 2020. URL: https://www.aclweb.org/anthology/2020.osact-1.0.

[8] P. Fortuna, S. Nunes, "a survey on automatic detection of hate speech in text", ACM Computing Surveys 51 (2018) 85.

[9] B. Roß, M. Rist, G. Carbonell, B. Cabrera, N. Kurowsky, M. M. Wojatzki, Measuring the Reliability of Hate Speech Annotations: The Case of the European Refugee Crisis, Originally published in Bochumer Linguistische Arbeitsberichte 17, NLP4CMC III: 3rd Workshop on Natural Language Processing for Computer-Mediated Communication, by Michael Beißwenger, Michael Wojatzki and Torsten Zesch (Eds.), 22 September 2016 (ISSN 2190-0949). (2016) 6–9.

[10] Z. Waseem, Are You a Racist or Am I Seeing Things? Annotator Influence on Hate Speech Detection on Twitter., in: Proceedings of the First Workshop on NLP and Computational Social Science, 2016, pp. 138–142.

[11] T. Mandl, S. Modha, P. Majumder, D. Patel, M. Dave, C. Mandlia, A. Patel, Overview of the hasoc track at fire 2019: Hate speech and offensive content identification in indo-european languages, in: Proceedings of the 11th Forum for Information Retrieval Evaluation, FIRE '19, Association for Computing Machinery, New York, NY, USA, 2019, p. 14–17. doi:10.1145/3368567.3368584.

[12] V. Basile, C. Bosco, E. Fersini, D. Nozza, V. Patti, F. M. Rangel Pardo, P. Rosso, M. Sanguinetti, SemEval-2019 task 5: Multilingual detection of hate speech against immigrants and women in Twitter, in: Proceedings of the 13th International Workshop on Semantic Evaluation, Association for Computational Linguistics, Minneapolis, Minnesota, USA, 2019, pp. 54–63. URL: https://www.aclweb.org/anthology/S19-2007. doi:10.18653/v1/S19-2007.

[13] J. M. Struß, M. Siegel, J. Ruppenhofer, M. Wiegand, M. Klenner, Overview of GermEval Task 2, 2019 Shared Task on the Identification of Offensive Language, in: G. S. for Computational Linguistics (Ed.), Proceedings of the 15th Conference on Natural Language Processing (KONVENS) 2019, s.a., Nürnberg/Erlangen, 2019, pp. 354–365. doi:10.5167/uzh-178687.

[14] Y. LeCun, L. Bottou, Y. Bengio, P. Haffner, Gradient-based learning applied to document recognition, in: Proceedings of the IEEE, 1998, pp. 2278–2324.

[15] Y.-L. Chung, E. Kuzmenko, S. S. Tekiroglu, M. Guerini, CONAN - COunter NArratives through nichesourcing: a multilingual dataset of responses to fight online hate speech, in: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, Florence, Italy, 2019, pp. 2819–2829. URL: https://www.aclweb.org/anthology/P19-1271. doi:10.18653/v1/P19-1271.

[16] T. Davidson, D. Warmsley, M. Macy, I. Weber, Automated hate speech detection and the problem of offensive language, in: Proceedings of the 11th International AAAI Conference on Web and Social Media, ICWSM '17, 2017, pp. 512–515.

[17] D. P. Kingma, J. Ba, Adam: A method for stochastic optimization, in: Y. Bengio, Y. LeCun (Eds.), 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings, 2015. URL: http://arxiv.org/abs/1412.6980.

[18] P. Kralj Novak, J. Smailović, B. Sluban, I. Mozetič, Sentiment of emojis, PLOS ONE 10 (2015) e0144296. doi:10.1371/journal.pone.0144296.

[19] J. Bevendorff, B. Chulvi, G. L. D. L. P. Sarracén, M. Kestemont, E. Manjavacas, I. Markov, M. Mayerl, M. Potthast, F. Rangel, P. Rosso, E. Stamatatos, B. Stein, M. Wiegmann, M. Wolska, , E. Zangerle, Overview of PAN 2021: Authorship Verification,Profiling Hate Speech Spreaders on Twitter,and Style Change Detection, in: 12th International Conference of the CLEF Association (CLEF 2021), Springer, 2021.

[20] M. Potthast, T. Gollub, M. Wiegmann, B. Stein, TIRA Integrated Research Architecture, in: N. Ferro, C. Peters (Eds.), Information Retrieval Evaluation in a Changing World, The Information Retrieval Series, Springer, Berlin/Heidelberg/New York, 2019, pp. 123–160. doi:10.1007/978-3-030-22948-1_5.