Introducing an Abusive Language Classification Framework for Telegram to Investigate the German Hater Community

Maximilian Wich, Adrian Gorniak, Tobias Eder
Daniel Bartmann, Burak Enes Çakici, Georg Groh
Technical University of Munich, Munich, Germany
{maximilian.wich, adrian.gorniak, burak-enes.cakc}@tum.de,
{tobias.eder, daniel.bartmann, grohg}@in.tum.de

Abstract

Because traditional social media platforms continue to ban actors spreading hate speech or other forms of abusive languages (a process known as deplatforming), these actors migrate to alternative platforms that do not moderate user content to the same degree. One popular platform relevant for the German community is Telegram for which limited research efforts have been made so far.

This study aimed to develop a broad framework comprising (i) an abusive language classification model for German Telegram messages and (ii) a classification model for the hatefulness of Telegram channels. For the first part, we use existing abusive language datasets containing posts from other platforms to develop our classification models. For the channel classification model, we develop a method that combines channel-specific content information collected from a topic model with a social graph to predict the hatefulness of channels. Furthermore, we complement these two approaches for hate speech detection with insightful results on the evolution of German speaking communities focused on hateful content on the Telegram platform. We also propose methods for conducting scalable network analyses for social media platforms to the hate speech research community. As an additional output of this study, we provide an annotated abusive language dataset containing 1,149 annotated Telegram messages.

Introduction

Hate speech and other forms of abusive language are a severe challenge that social media platforms, such as Facebook, Twitter, and YouTube, are facing nowadays (Duggan 2017). Moreover, this problem is not only limited to the online world; studies have shown that online hate correlates with physical crimes in the real world (Müller and Schwarz 2021; Williams et al. 2020), making the phenomenon a societal challenge for everybody.

To enforce a fast reaction to harmful content on social media platforms, Germany has passed a set of laws (Network Enforcement Act or NetzDG) to force social media companies to take action against hate speech on their platforms (Rafael 2019; Echikson and Knodt 2018). These actions range from deleting single posts that contain hateful content to banning actors from the platform, which is called deplatforming (Fielitz and Schwarz 2020). While deplatforming helps limit the reach of these hate actors (Fielitz and Schwarz 2020), it often makes them migrate to less or un-regulated platforms and continue their hateful communication (Rogers 2020; Fielitz and Schwarz 2020; Urman and Katz 2020); one such alternative social media platform is Telegram (Rogers 2020; Fielitz and Schwarz 2020; Urman and Katz 2020). In Germany, Telegram has become the focal point for right-wing extremists, conspiracy theorists, and COVID-19 deniers (Fielitz and Schwarz 2020; Urman and Katz 2020; Eckert, Leipertz, and Schmidt 2021). Along with this rapid increase in popularity and usage by various user types, two important challenges regarding abusive language detection arise: first, the automatic detection of abusive content in such texts and, second, an aggregated view on the account level to identify hateful accounts. For both challenges, we propose a machine learning-based approach.

Previously, most research efforts on detecting hate speech focused on posts and comments from Twitter and Facebook (Ross et al. 2016; Bretschneider and Peters 2017; Struß et al. 2019; Wiegand, Siegel, and Ruppenhofer 2018; Mandl et al. 2019, 2020; Wich, Räther, and Groh 2021; Wich et al. 2021a) with very little focus on Telegram. This is particularly the case for content in German. At the same time, Telegram channels and chat groups are known for being a prime driving factor of online hate within the German language community. We want to bridge this gap and build abusive language classification models for Telegram messages. Because there is no abusive language dataset available that contains labeled Telegram messages in German, our approach is to use existing abusive language datasets in German, collected from other platforms and construct a classification model for Telegram. This leads to the first research question for this study:

RQ1 Can existing abusive language datasets from other platforms be used to develop an abusive language classification model for Telegram messages?

Because the development of an abusive language classification model requires significant amounts of data, we collected such data from the platform (Telegram) over a longer period of time. By collecting the data, we are also able to formulate additional questions about the type of content and its spread on the platform. Because there is little research on these types of communication channels and their content, we were also interested in how this content has changed over the observed time period, during which deplatforming was occurring on other social media. Thus, we formulate an additional research question in terms of message contents:

RQ2 How did the prevalence of abusive content evolve in the last years on Telegram?

Moving away from the message-level approach and towards an user-based approach for abusive language detection, so far no methodology has been introduced to address this problem for Telegram. As a solution, we propose developing a graph model leveraging topical information for channels in the German hater community on Telegram to find suitable representations, leading to the third research question:

RQ3 Can a classification model be used to predict whether a Telegram channel is hateful or not?

Lastly, maintaining the channel perspective, we were interested to investigate whether our approach would allow for the derivation of channel clusters and communities, which is another important aspect regarding online hate. For this, we analyzed the topical distribution and the graph embeddings for each channel, resulting in research question four:

RQ4 Can we leverage the topical distribution and graph embeddings to derive meaningful clusters from channels?

As an additional contribution, we release an abusive language dataset containing 1,149 Telegram messages labeled as *abusive* or *neutral*. [1]

Related Work

Studies on Telegram are limited, but the number began to grow in the past years. Baumgartner et al. (2020) released an unlabeled dataset containing 317,224,715 Telegram messages from 27,801 channels, which were posted between 2015 and 2019. They used a snowball sampling strategy to discover channels and collect messages, starting with approximately 250 seed channels (mainly right-wing channels or channels about cryptocurrency). Rogers (2020) conducted an empirical study on actors who were deplatformed on traditional social media and migrated to Telegram. As part of their study, they used a classification model based on hatebase.org to detect messages with hateful language (Rogers 2020). Previous studies on the platform Twitter have shown that identifying networks and user context for social media have significant beneficial impact on classification tasks, such as hate speech detection (Mosca, Wich, and Groh 2021; Wich et al. 2021b) and motivate further in-depth studies on these communities on other platforms. Urman and Katz (2020) conducted an in-depth network analysis of a far-right community on Telegram. They used a snowball sampling strategy to uncover this community, starting with a German-speaking far-right actor. Fielitz and Schwarz (2020) analyzed German hate actors across various social media platforms and investigated the impact of deplatforming activities on these actors. According to them, ”Telegram has become the most important online platform for hate actors in Germany” (Fielitz and Schwarz 2020, p. 5). With a focus on COVID-19, Hohlfeld et al. (2021) and Holzer (2021) investigated public German-speaking channels on Telegram. The only labeled abusive language dataset with Telegram messages that we found is provided by Solopova, Scheffler, and Popa-Wyatt (2021). They released a dataset containing 26,431 messages in English from a channel supporting Donald Trump. To the best of our knowledge, no study has developed an abusive language classification model for German Telegram messages or channels.

Because there is no annotated German Telegram dataset available, we decided to train our classification model on existing German abusive language datasets. In total, we found eight of such datasets (Ross et al. 2016; Bretschneider and Peters 2017; Wiegand, Siegel, and Ruppenhofer 2018; Struß et al. 2019; Mandl et al. 2019, 2020; Wich et al. 2021a; Wich, Räther, and Groh 2021). We decided to use five of them—which constitute the most recent ones, excluding Wich et al. (2021a). These five datasets have comparable label schemata, and a large portion of the data is from the same period as our collected Telegram data. Wich et al. (2021a) was excluded because their data were only pseudo-labeled. More details on the selected datasets can be found in the following section.

As part of our methodology we worked with the Perspective API[2] to classify subsets of messages from Telegram for our semi-supervised baseline comparison. Recent studies that also dealt with the Perspective API have shown systemic bias in their classification framework, which could lead to minority groups being overly flagged by such hate speech systems (Sap et al. 2019, 2021). Sen et al. (2021) similarly performed a study on the Perspective API to discuss potential scientific pitfalls with the usage of automated classification for the social sciences. In our case this problem is dampened firstly by the clear focus on German language text, in which minority German speaker's vernacular is not as pronounced or flagged as offensive speech, but moreover secondly by our sampling strategy, which aims to capture right-wing hate groups and their networks. Through this we are interested in determining the potential toxicity of a very specific subgroup, which in the past was deplatformed for reasons of toxicity and hate speech already. Still we are aware of the limitations of a semi-supervised approach and further studies of the matter should verify results by including domain experts, such as anti-hate speech groups and activists.

Methodology

In the first part, we describe how we collected data from Telegram. After that, we provide details on how we developed the abusive classification model for Telegram messages based on datasets from other platforms. In the third part, we describe how we developed a classification model to predict

[1] Code and data available on GitHub:
https://github.com/mawic/telegram-abusive-language-classification

[2] https://www.perspectiveapi.com/

whether a channel belong to the hate category based on the results from the message classifier and the social graph.

Collecting Data

We used a snowball sampling strategy to collect data from Telegram. We only collected messages from public channels that were accessible via the website t.me. A channel is comparable to a news feed: the channel operator can broadcast messages to subscribers of the channel, but subscribers cannot directly post messages on the channel. Groups and private chats were excluded from the data collection process. As seeds for the snowball sampling strategy, we used a list of German hate actors proposed by Fielitz and Schwarz (2020). At the time of data collection, 51 channels from Fielitz and Schwarz (2020)'s list were still accessible. The list comprises, among others, far-right actors, supporters of Qanon, and alternative media.

In the first round of snowball sampling, we collected messages from all seed channels. In the next round, we collected all channels that were mentioned in messages collected from the first round or whose messages were forwarded by the channels of the first round. We repeated this procedure in the third round, but we excluded some of the newly discovered channels due to the large number of channels. We defined a threshold: a channel must be mentioned or forwarded by at least five channels to collect its messages. From all channels, we collected messages that were posted between 01/01/2019 and 03/15/2021.

After data collection, we conducted language detection on the messages because the crawling process also collects other language channels such as Russian and English and we wanted to keep the focus on German. We used multilingual word vectors from *fastText* to classify languages (Grave et al. 2018). The language detection here is based on the message text and a link preview if it exists. In a second step, the language labels of messages are aggregated on a channel level. The language of a channel is German if it is the most or second most common language in the channel. The reason for the latter is that some German channels primarily share content from foreign-language sources. In the following sections, all results refer to the German-speaking channels of the dataset.

Building Classification Models for Telegram Messages

Models To classify Telegram messages, we trained several binary classification models on different German datasets. The goal is to combine multiple classifiers to improve classification performance because each dataset covers different aspects and topics of abusive languages. The reason for focusing on binary classification was that it makes combining classifiers easier.

All classification models are based on pretrained BERT base models (Devlin et al. 2019). We used `deepset/gbert-base` (Chan, Schweter, and Möller 2020) and `dbmdz/bert-base-german-cased`[3] depending on the model's performance on the individual dataset. Our hyperparameters for training comprise a maximum number of eight epochs, a learning rate of 5×10^{-5}, and a batch size of eight. In addition, we implemented an early stopping callback that stops the training after four consecutive epochs without any improvement. We selected the model with the highest macro F1 score on the validation set.

Before training the models, texts are preprocessed. The preprocessing steps comprise, among others, masking URLs and user names and replacing emojis.

Data We used the following German abusive language datasets collected from different platforms (mainly Twitter) to train our models:

- *GermEval 2018*: Wiegand, Siegel, and Ruppenhofer (2018) released an offensive language dataset as part of the shared task GermEval Task 2018. It contains 8,541 tweets with a binary label (*offense*, *other*) and a fine-grained label (*profanity, insult, abuse, other*). We used the train/test split proposed by the authors and used a 90/10 split for the training/validation set.
- *GermEval 2019*: Struß et al. (2019) published an offensive language dataset that is part of the GermEval Task 2019. It comprises 7,025 tweets that are labeled with the same labeling schema, as the previous dataset, but a further dimension was added *(implicit, explicit*). The data were split in the same way as GermEval 2018.
- *HASOC 2019*: Mandl et al. (2019) released a multilingual hate speech and offensive language dataset, called "Hate Speech and Offensive Content Identification in Indo-European Languages" (Mandl et al. 2019, p. 1), as part of a shared task. It comprises posts from Facebook and Twitter in German, English, and Hindi. The German part comprises 4,669 records with a binary label (*non hate-offensive, hate and offensive*) and a fine-grained label (*hate*, *offensive*, *profanity*). We used the train/test split proposed by the authors and used a 90/10 split for the training/validation set.
- *HASOC 2020*: Mandl et al. (2020) published another dataset, which is comparable to the previous one. It consists of posts from YouTube and Twitter in German, English, and Hindi. The German part has a size of 3,425 records using the same labeling schema as the previous dataset. We used the proposed train/validation/test-split of 70%/15%/15%.
- *COVID-19*: Wich, Räther, and Groh (2021) released an abusive language dataset containing 4,960 German tweets that primarily focus on COVID-19. These tweets have a binary label (*neutral*, *abusive*). We used a train/validation/test split of 70%/15%/15%.

We trained individual classification models for all datasets, except for HASOC 2019 because we could not train a model that provides an acceptable classification performance. Furthermore, we combined the GermEval and HASOC datasets and trained two additional classifiers on the two combined datasets. Combining these datasets was possible because the respective datasets use the same labeling schema.

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

Classifying Telegram Messages Because a Telegram message can have up to 40,986 characters, the tokenized message may exceed the maximum sequence length of the BERT model, which is 512. To tackle this problem, we split all messages that had more than 412 words into parts with a maximum length of 412 words. When splitting a message, we made sure not to split sentences. For this purpose, we used the sentence detection method of the library spaCy (Honnibal et al. 2020). There were two reasons for setting the threshold to 412 words. First, using words instead of tokens was easier during preprocessing. Second, a word can be tokenized into multiple tokens. Therefore, we set the threshold to 412 instead of 512. Every part of the split message was individually classified. The final label of the complete message results from the highest probability for the abusive class. The reason for this approach was because an abusive text can contain nonabusive parts but not the other way around. In addition to the six classification models, we used Googles Perspective API to classify the same messages. The API returns a toxicity score between 0 and 1, representing the likelihood that a message should be considered as toxic. Additionally the API offers several models for other factors such as identity attack, insult, profanity, threat etc. In our study we chose general toxicity as the most universal label. While this includes examples like profanity, which are not strictly hate speech related, we chose the broader perspective to represent the extent of flagged content in the network. We used these general toxicity classification results as a semi-supervised baseline to benchmark our models.

Evaluating Classification Models To evaluate the classification performance of our trained models on Telegram messages, five annotators manually annotated 1,150 of the classified Telegram messages. More information about the annotators follows below. The 1,150 messages originated from two different sampling strategies. The first strategy uses the classification results of the six trained models and the Perspective API. For each classifier, we sampled 50 messages classified as abusive and 50 classified as neutral, resulting in a total of 700. The second strategy used a topic model trained on Telegram messages (more details on the topic model can be found in the subsection Topic Model). We randomly sampled 30 messages from the 15 most prominent topics. Finally, we ensured that the annotation candidates do not contain any duplicates. As a result, we assured that the dataset has a certain degree of abusive content and that it represents the most relevant topics.

We use the labeling schema of the *COVID-19* dataset proposed by Räther (2021) and Wich, Räther, and Groh (2021) because it is compatible with the binary schema of the *HASOC* and *GermEval* datasets:

- *ABUSIVE*: The tweet comprised "any form of insult, harassment, hate, degradation, identity attack, or the threat of violence targeting an individual or a group. " (Räther 2021, p. 36)
- *NEUTRAL*: The tweet did "not fall into the *ABUSIVE* class." (Räther 2021, p. 36)

Data were annotated by four nonexperts and one expert, who are males and in their twenties or early thirties. The annotation process consisted of three phases. In phase 1, the expert presented and explained the annotation guidelines to the four nonexperts. Subsequently, all five annotators annotated the same 50 messages. In 18 cases, the annotators did not agree on the final label. These cases were discussed in a meeting to align the five annotators. In phase 2, the annotators annotated the remainder of the 1,150 messages. Each message was annotated by two different annotators. The annotators were allowed to skip a message if they could not decide on a label. In phase 3, messages without a consensus were annotated by three additional annotators so that a majority vote was possible. We used Krippendorffs alpha (Krippendorff 2004) to measure inter-rater reliability. To assist in annotations, we used the text annotation tool of Kili Technology (Kili Technology 2021).

Combining Classification Models Because the datasets and consequently the classification models cover different aspects of abusive languages, we combined the six classifiers to improve classification performance (Perspective API was not part of the combination). The labels produced by this combination were used for subsequent experiments.

Analyzing Evolution of Abusive Content We performed two analyses to evaluate the evolution of abusive content in the German hater community on Telegram to answer RQ2. First, we compared the number of abusive messages with all messages from the collected German channels between 01/01/2019 and 02/28/2021 on a monthly level. We excluded the messages posted in March 2021 because we did not have data for the entire month. Then, we examined the relative share (prevalence) of abusive content in the messages from all German channels for the same period and granularity. In addition, we reported the prevalence of abusive content from the seed channels and the 1st-degree network of the seed channels.

Building a Classification Model for Hatefulness of Channels

Channel Labels We chose to frame the task as a classification problem deciding on a binary choice of *hater* and *neutral* channels. This formulation was preferable over a formulation as a regression problem, which predicts the relative *hatefulness* of channels, due to the fact that even channels with the highest amount of hate content, still contain a vast amount of non-hate messages. The average portion of hate messages in the channels of the selected network is 2.7% with a standard deviation of 0.045. Similarly, we were more interested in mapping out the overall extent of the network sharing similar content, than to focus on hotspots based entirely on the intensity of the hate, as opposed to their centrality within the network. To set up the task we had to determine a label for each channel based on whether or not the channel contained any abusive messages. We at first defined a *hater* as a channel that posted or forwarded at least one abusive message. This minimum threshold is chosen based on the fact that we want to generate a comprehensive overview of the potential extent of the spread of hate content on the platform. While it is possible to set the bar for the hate label higher, we were primarily interested in all channels spreading this type of information and not just in the most prolific spreaders. At the same time, setting the threshold to one proved problematic due to the possibility of misclassification, meaning that false positives would cause neutral channels to be classified as haters. Instead, for each message, we calculated a threshold based on the conditional probability that a message is neutral under the condition of it being labeled as abusive. This conditional probability is retrieved from a confusion matrix (Figure 1h). As a result, we had to adjust the weighting of the confusion matrixs rows. Because we intentionally oversampled the abusive class in the evaluation set, the ratio of abusive texts was no longer representative of the entire dataset. We assume that the relative share of abusive content is 3.1% for 2020, based on the results from the analysis of the abusive contents evolution. The resulting conditional probability is 82.9%. Assuming an error rate of smaller than 5.0% , we need at least 17 messages that are classified as *abusive* to be certain that the abuse posted is likely to be genuine. Second, we created a directed graph representing the network of channels. Each channel is a node; a directed edge from nodes A to B exists if A either mentions B or forwards a message from B.

Topic Model We assigned a topic distribution vector as a feature to each node of the graph, representing the topical distribution within the messages of the channel. The topical distribution was calculated on the basis of the topic model generated with Top2Vec (Angelov 2020). We relied on the hyperparameter selection of the author, used the `distiluse-base-multilingual-cased`[4] pretrained sentence transformer as embedding model, and sampled 250,000 messages (500 messages from the 500 channels containing the largest amount of messages in our dataset) as training samples. From the 100 most relevant topics, we manually chose nine topics to serve as proxies for hateful content. These topics were predominant in a larger number of channels, while simultaneously being indicative of hatefulness, predominantly by being focused on a specific kind of discriminative or otherwise *abusive* language. They are listed in Table **??**: the topic name in the first column was derived on the basis of the most descriptive terms of the respective topic vectors from which we provide the first three terms in the second column (in German) and a translation of the terms in the third column. Because we are working with many channels that can be associated with German hater communities, we relied only on these topics to cluster different topical emphases with respect to potentially harmful content. We aggregated the counts of all documents in our dataset with cosine similarity to any of the selected topics greater than 0.5 and normalized these counts to create a topic distribution for each node.

Graph Model We used GraphSAGE to generate embeddings for the graph (Hamilton, Ying, and Leskovec 2017). The graph was the one described in the paragraph *Channel Labels* and combined with the topic distribution vectors as node attributes from the previous paragraph. We used the Directed GraphSAGE method from the StellarGraph library (CSIRO's Data61 2018). As we were learning unsupervised embeddings, i.e., we did not provide the learning model with labels of the channels, we used the *Corrupted Generator* of StellarGraph for sampling additional training data. During training, the model learned to differentiate between true graph instances and corrupted ones. The model was trained for 500 epochs with two layers of size 32 each, an Adam optimizer, and an early stop after 20 epochs of patience.

Channel Classification We developed a neural network (NN) classification model using the graph embeddings to predict the classes. The model consists of two densely connected NN layers. The input for the first layer is a 32-dimensional graph embedding. The second layer (output) has two units due to the binary task. The first layer uses a rectified linear unit activation function, whereas soft-max was applied to the output layer. To train the model, cross-entropy was used as a loss function with accuracy as the metric using an Adam optimizer. We trained the model for a maximum of 150 epochs with a batch size of eight with an early stopping strategy that had the patience of 100 epochs and a minimum delta of 0.05 for accuracy on the validation set. The dataset was split into training/validation/test sets (70%/15%/15%).

The dataset for RQ3 only used messages from 2020, as the social network on Telegram is rapidly evolving and changing, with channels and users not staying constant over longer periods of time. That means that by including older edges the overall network structure would generally be less meaningful and introduce noise into the analysis. Another aspect of this decision is that the emergence of COVID-19 strongly influenced and accelerated the evolution of the network, which did not exist pre-COVID-19 pandemic.

Results

Collecting Data

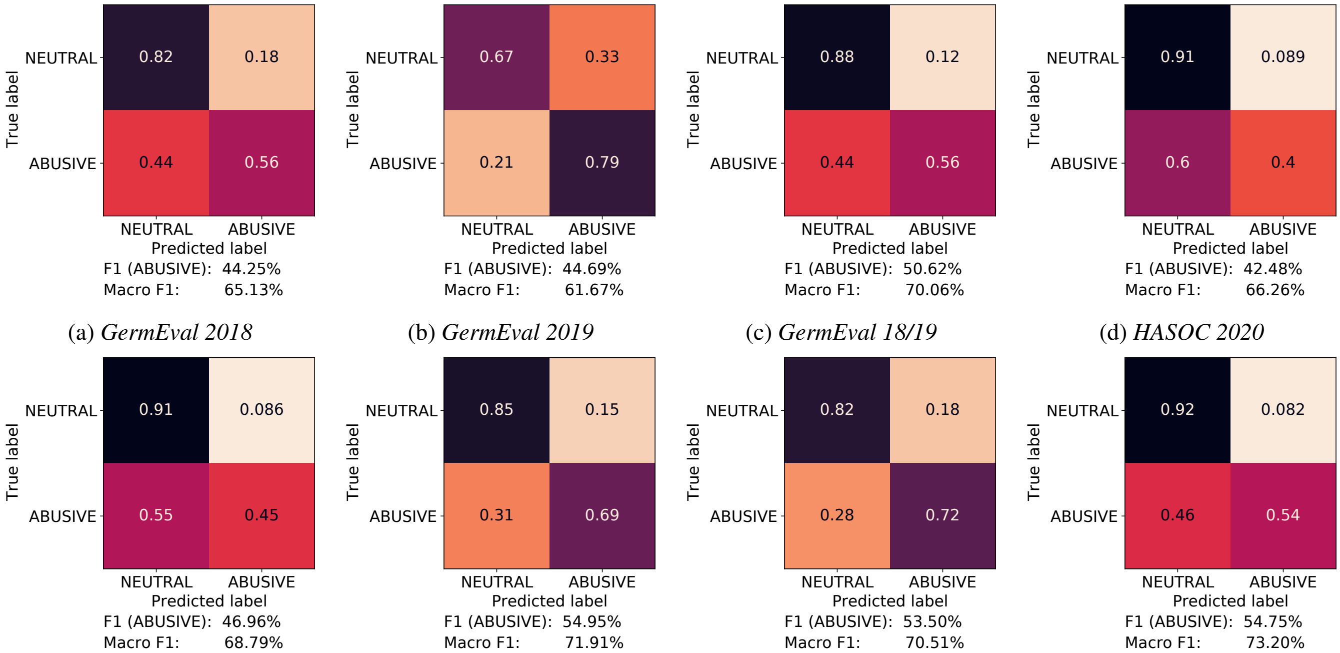
In total, we collected 13,822,605 messages from 4,962 channels that were posted between 01/01/2019 and 03/15/2021. 28.4% of all messages (3,931,136) are forwarded messages, showing the popularity and relevance of this feature for Telegram. In addition to the 4,962 channels, we collected the metadata of 43,142 additional channels that were either the source of forwarded messages or were mentioned in a message.

39.2% of all collected messages (5,421,845) are in German, which is the most frequent language, followed by English and Russian. 2,748 of the 4,962 crawled channels (55.4%) are classified as German-speaking according to our approach and are therefore included in the full analysis.

[4] https://huggingface.co/distilbert-base-multilingual-cased

Topic	Descriptive terms	Translation
Vaccinations	impfen, geimpft, durchgeimpft	vaccinate, vaccinated, fully vaccinated
Police	Polizeigewalt, Bundespolizei, Polizeiführung	police violence, federal police, police leadership
COVID-19	Coronakrise, Corona, Coronaleugner	corona crisis, corona, corona denier
Migration	Migrantengewalt, Migranten, Refugees	migrant violence, migrants, refugees
Extremism	rechtsextremer, rechtsextremen, rechtsextreme	far-right
Racism	Rassismus, rassistischer, rassistisch	racism, racist
Islamophobia	Moslemterror, Islamisten, Islamisierung	Muslim terror, Islamists, Islamization
Violence	sterben, Massenmörder, Massenmord	die, mass murder
Antisemitism	Antisemismus, Antisemiten, antisemitische	antisemism, antisemites, antisemitic

Table 1: Topics selected for topic distribution along with three descriptive terms of the topic model.



(a) *GermEval 2018* (b) *GermEval 2019* (c) *GermEval 18/19* (d) *HASOC 2020*

(e) *HASOC 19/20* (f) *COVID-19* (g) *Perspective API (baseline)* (h) *Best combination*

Figure 1: Classification performance of the various models on annotated Telegram evaluation set.

Dataset/Model	Prec	Rec	F1	Macro F1	Basis
GermEval 18	71.1	61.0	65.7	75.0	dbmdz
GermEval 19	72.2	85.1	78.1	77.1	dbmdz
GermEval 18/19	87.6	77.6	82.3	83.8	dbmdz
HASOC 20	69.0	73.7	71.3	80.6	deepset
HASOC 19/20	71.0	69.9	70.4	80.3	dbmdz
COVID-19	73.9	69.9	71.8	82.3	deepset

Table 2: Classification performance of the classifiers

Building Classification Models for Telegram Messages

Models Table **??** presents the classification metrics of the six trained classification models. It comprises the precision, recall, and F1 score of the abusive class as well as the macro F1 score and the used model that performed best on the dataset. The last column contains the name of the pretrained model that was used as basis for fine-tuning.

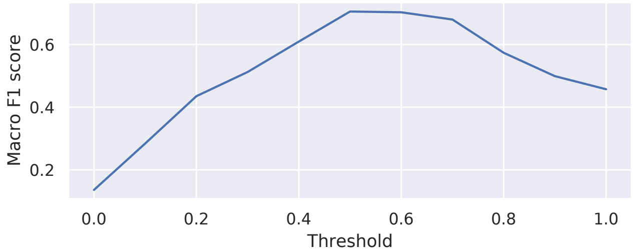
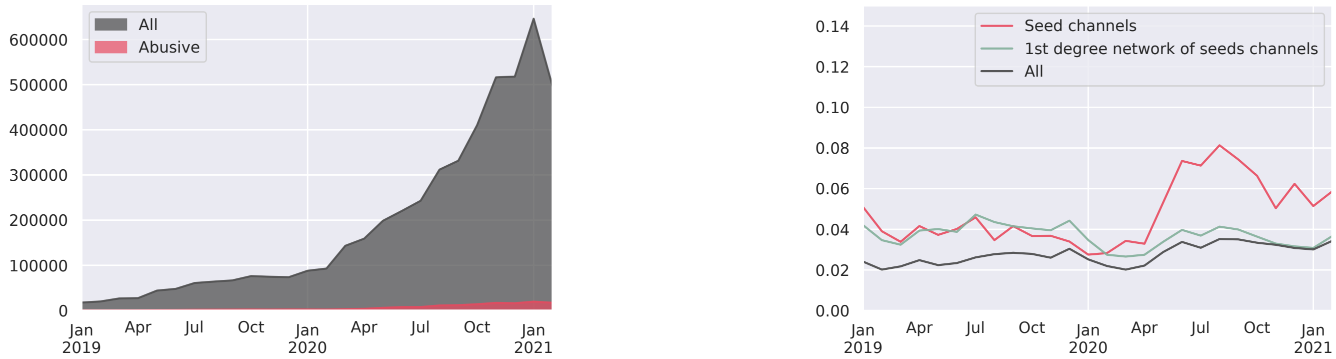


Figure 2: Macro F1 score dependent on various threshold for Perspective API on test set.

Evaluating Classification Models To test the trained classification models, we annotated 1,150 Telegram messages. One message was removed during the annotation process because it did not contain any text, resulting in 1,149 annotated messages. 968 (84.2%) were labeled as *neutral* and 126



(a) *Absolute number of all and abusive messages from German channels.*

(b) *Relative share of abusive messages for German channels.*

Figure 3: Evolution of abusive messages in absolute and relative terms.

(15.8%) as *abusive*. The Krippendorff's alpha was 73.87%, which is a good inter-rater reliability score in the context of hate speech and abusive language (Kurrek, Saleem, and Ruths 2020).

Figure 1 visualizes the classification performance of the various classifiers on the evaluation set. It presents the confusion matrix, the F1 score of the abusive class, and the macro F1 score of the six trained classification models (a–f), the Perspective API (g), and the best combination of the six classifiers (h). Let us first compare the six classification models that we trained on the different datasets. The best-performing model is COVID-19; it outperformed the other models in terms of F1 score (54.95%) and macro F1 score (71.91%). In comparison to the COVID-19 test set, however, the performance drastically decreased. This should not be surprising because Telegram messages differ from tweets in terms of structure and content.

To benchmark the performance of our classification model, we used Googles Perspective API to classify messages. The API returns a toxicity score between 0 and 1 which represents the probability of the message being toxic. We translated this value by setting a threshold. If the value is above or equal to the threshold, the label is *abusive*; otherwise, the label is *neutral*. We initially set the threshold for abusive messages to 0.5. Results after validation of other thresholds are collected in Figure 2; the highest macro F1 score on the test set is also achieved by setting a threshold of 0.5. Comparing the performance of the Perspective API with our best-performing model, our model achieves a slightly higher F1 score (54.95% vs. 53.50%) and macro F1 score (71.91% vs 70.51%) in the case of the chosen threshold. The model also achieves comparable results with slightly higher thresholds, with increasing decay in performance for higher toxicity scores, as more messages fall into the false positive category.

Because the datasets cover different aspects of abusive language, we also examined whether a combination of all six classifiers can improve performance. Performance indicates that a majority vote (at least four classifiers vote for *abusive*) of all six models is the best-performing combination in terms of the macro F1 score, as shown in Figure 1h. It outperforms the Perspective API and the classifier trained on the COVID-19 datatset in terms of macro F1 score. To validate the result, we applied the McNemar's test (Dietterich 1998) to show that the best combination performs significantly differently ($p < 0.05$) from the Perspective API ($p = 2.69 \times 10^{-5}$) and COVID-19 ($p = 1.02 \times 10^{-3}$). Therefore, the best combination is the majority vote with at least four classifiers voting for *abusive*, which we used for the following two case studies.

Analyzing the Evolution of Abusive Content Figure 3a shows how the number of messages in the German Telegram channels has increased between the beginning of 2019 and 2021. We can trace the growth of these channels back to the phenomenon of deplatforming. Deplatforming means that actors are permanently banned on traditional social media platforms (e.g., Facebook, Twitter, and YouTube), resulting in them moving to less moderated or unregulated platforms (e.g., Telegram and Gab) (Rogers 2020; Fielitz and Schwarz 2020; Urman and Katz 2020). Notably, the increase in messages accelerated substantially with the rise of the COVID-19 pandemic (February 2020). The reasons for this are likely similar. Traditional social media platforms (e.g., Twitter and YouTube) blocked accounts of hate actors spreading conspiracy theories regarding COVID-19, causing migration to Telegram and alternative platforms (Fielitz and Schwarz 2020; Holzer 2021). Simultaneously with the growing number of messages every month (black curve), abusive content also increased (red curve).

To answer the question of whether the abusive content has grown only proportionally, we plotted the relative share of abusive content in Figure 3b. The black line represents the relative share for all messages. We observe that the share of abusive content increased from 2.4% to 3.4% during the 26 months. The red line shows the portion of abusive messages in the seed channels. It is not surprising that the share is significantly higher because these channels were labeled as hater channels by Fielitz and Schwarz (2020). The line follows the trend: the abusive content of the selected channels is growing. The green line visualizing the percentage of abusive messages in the channels being in the first-degree network of the seed channels[5] does not mirror the same trend. A potential explanation is that the number of channels in the first-degree network has increased over time, causing the alignment of the relative share with the overall average. In total, the prevalence of abusive content for the entire period is 3.1% for all channels, 5.3% for the seed channels, and 3.5% for the 1st degree network of the seed channels.

In summary, we observe the trend that messages classified as *abusive* by our combined model increase in absolute and relative terms in the German hater community on Telegram and are particularly pervasive in the central seed channels.

Building a Classification Model for the Hatefulness of Channels

In this section, we report the results of our classification model for identifying hateful users, along with additional findings in the process of setting up our model.

Channel Labels The dataset for developing a channel classification model contains 2,420 German channels that were active in 2020 and posted 3,232,721 messages. 809 of 2,420 channels (33.4%) are labeled as *hater*, the rest as *neutral*. Each channel is represented by a node in the directed graph. In total, we identified 146,865 edges between channels, which represent messages from one channel which are shared in another or mentioning another channel in a message (unidirectional). This leads to a density of 0.0251 and an average in- and out-degree of 60.73.

Topical Distribution As the first result, we examined clusters based on the topical distribution of the seed channels. To do this, the similarity between the topical distribution of each pair of users has been computed using the Jensen-Shannon divergence. For the resulting similarity matrix, a hierarchical clustering approach has been used to group similar users into clusters, as described in Figure 4. While we only disclose an anonymized version of our results, we report that the upper left cluster consists only of sources for alternative news and the large cluster in the center mainly contains actors who belong to the far-right network.

Graph Embeddings Before using the graph embeddings from the directed GraphSAGE model for the classification model, we investigated the expressiveness of the embeddings for community detection. For this, we applied the dimensional reduction method UMAP to our embeddings to find more dense representations. In the second step, we used DB-SCAN to cluster these reduced embeddings. In Figure 5, we report the results of the community detection, along with a visualization indicating the label of each node (channel). Seed channels are marked with a large square instead of a dot. The clustering algorithm recognizes four distinct communities along with one outlier class. The channels/nodes of the outlier class are dark green and spread over the figure (cf. Figure 5a). The large community in the center does not only contain most of the seed channels in our dataset but also the largest proportion of channels labeled as *hater* (38%). In the other communities, we find a significantly lower proportion of hatefully classified users (5%-24%). In the outlier class, 33% are hater. From that, we conclude that hateful users appear more often in communities with other hateful users.

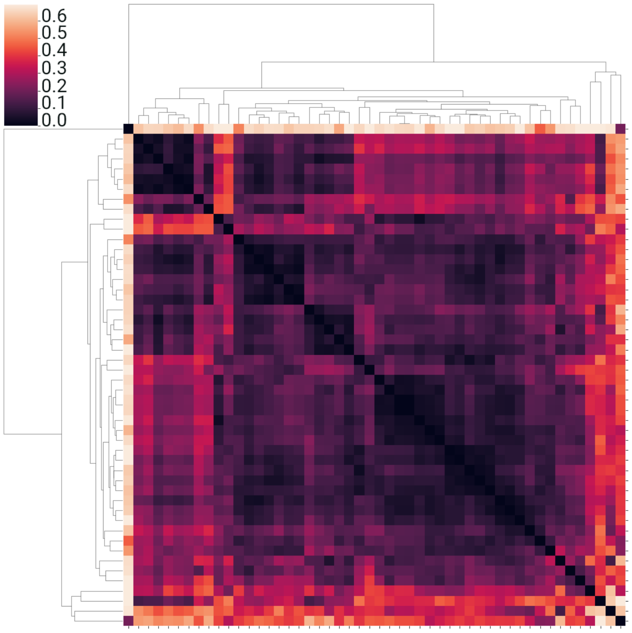


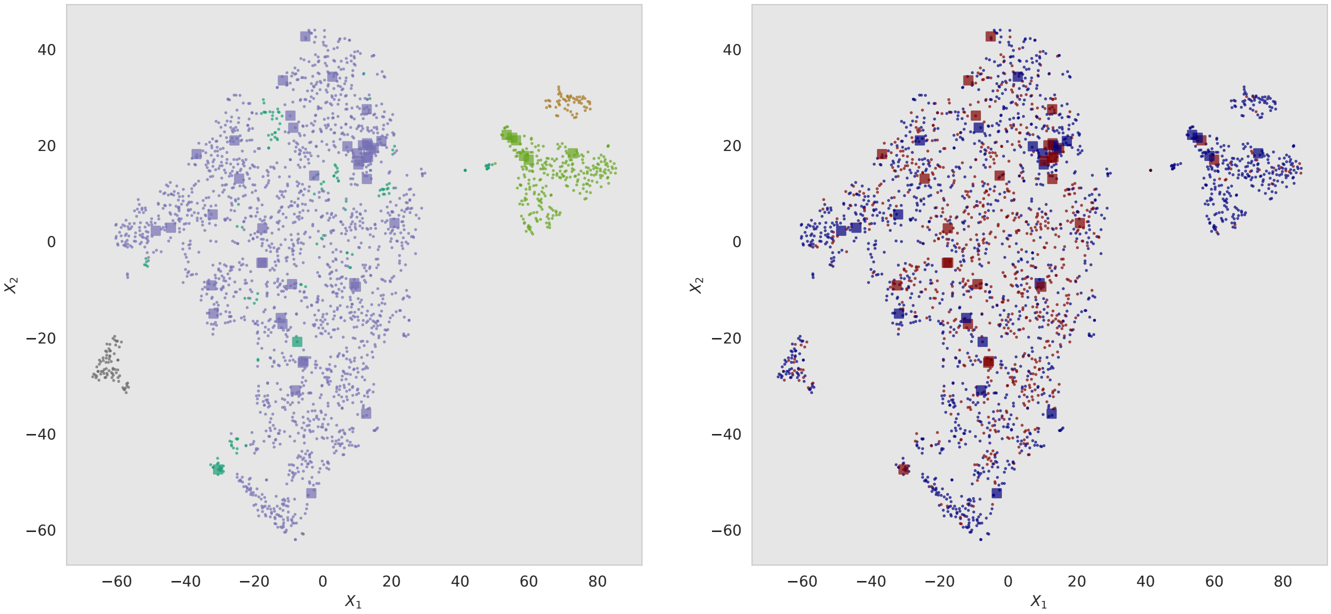
Figure 4: Similarity matrix for the seed channels of the Telegram dataset. A detailed list of channels can be found in the resources on Github, see above.

Channel Classification The classification model trained to distinguish between *hater* and *neutral* channels achieves a macro F1 score of 69.5% (*neutral*: 74.2%; *hater*: 64.9%). It is important to stress that this performance is reached solely on the unsupervised graph embeddings as input and does not use any additional semantic or text data. Figure 6 visualizes the confusion matrix of the classification model for the test set. We observe that the model performs well in predicting the labels of the German Telegram channels.

Discussion

In RQ1 we asked whether existing abusive language datasets can be used to train language classification models for the Telegram platform. The short answer to this is yes. However, we have to accept a decline in classification performance. Comparing the macro F1 scores of the classifiers on the original test and evaluation sets, we observe an average decline of approximately 12.5pp. To better assess this value, it is helpful to look into the study on the generalizability of abusive language datasets from Swamy, Jamatia, and Gambäck

[5] A channel is in the first-degree network if a seed channel mentions the channel or forwards a message from this channel and vice versa.



(a) *Graph embeddings with community labels*

(b) *Graph embeddings with hate class labels (red=haters)*

Figure 5: Comparison of graph embeddings with community and hate class labels.

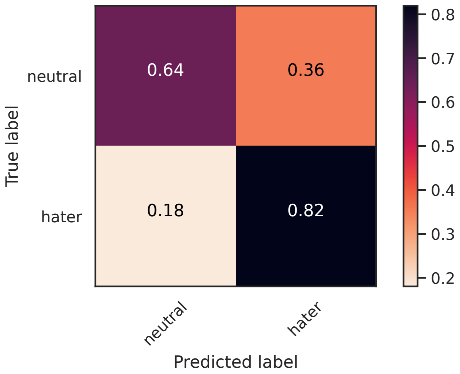


Figure 6: Confusion matrix of model to classify channels.

(2019). They trained models on different abusive language datasets and evaluated them on each other. The average performance decline is 18.1pp if a classifier is evaluated on another test set. Considering this aspect, we can claim that our models perform decently, especially the combination of all six classification models with a threshold of four. This claim is supported by the fact that the combined models outperform the Perspective API in terms of F1 score. We integrated this external model provided by Google as a benchmark because it is developed to handle different types of texts (e.g., comments, posts, and emails), and it is in production (Jigsaw 2021). Further, utilizing Twitter abusive language data has proven as particularly helpful in this case, as it offers the largest amount of labeled datasets in German currently available. The availability of the data is not easily compensated by a smaller labeled dataset entirely focused on Telegram, or other social media platforms, such as Reddit. In the end, while it is to be expected that platform specific data would be beneficial for better performance on the task, the core idea of the research was also about tracing the effects of deplatforming and the shift from one social media platform, such as Twitter, to another. It is to be expected that despite the changing particularities on Telegram, deplatformed actors would still choose to communicate in a similar manner as on the previous platform and talk about similar topics. Further the experiments are also fruitful in case additional platforms, such as Telegram would in the future choose to deplatform certain actors, in which case data would need to be collected from the ground up again. Consequently, we can state that our approach is successful, but it still provides room for improvement.

In RQ2 we wanted to observe the changes in abusive content on Telegram over the deplatforming period on other social media sites. We observe an increasing prevalence of abusive messages in the collected Telegram subnetwork, especially in the group of the seed channels. Notably, the rise of COVID-19 coincided with a significant increase of abusive messages. One may argue that the absolute share of abusive content is unreliable because our combined classification model is imperfect. However, the observed change in the relative share of abusive messages provides a reliable indication of an increasing amount of overall abusive content since it was classified using the same classification model. We trace this trend back to the deplatforming activities of large social media platforms and Telegrams lack of content moderation. However we also have to point out that the prevalence of abusive content is unrepresentative of the entire German Telegram network. Due to our snowball sampling approach, we have an obvious selection bias because we started with channels that were classified as hate actors by Fielitz and Schwarz (2020). Nevertheless, we assume that the prevalence of abusive content is larger on Telegram than on traditional social media platforms, such as Twitter, Facebook, and YouTube, that have implemented rigorous reporting and monitoring processes and take an active stance in content moderation. In the case of Telegram, such processes are missing, or entirely in the hands of the respective channel owners.

In RQ3 we asked whether classifying hateful content on a channel level was possible using only aggregate information and the overall network structure. We thus developed a classification model to predict whether a channel is a hate actor. It uses the network structure and the topic distribution of messages in each channel for prediction. Our model achieves a macro F1 score of 69.5%. To the best of our knowledge, we are the first to develop such a classification model for Telegram channels. Therefore, we do not have a baseline to compare our results with. However, Ribeiro et al. (2018) and Li et al. (2021) developed comparable models classifying Twitter accounts as hateful or normal. For the same dataset, Ribeiro et al. (2018) and Li et al. (2021) achieved F1 scores of 67.0% and 79.9%, respectively. Our F1 score of 64.9% is not directly comparable with these results, but it is in a similar order of magnitude, supporting our approach.

In RQ4 we wanted to find out whether we can leverage topical distributions combined with graph embeddings, to derive meaningful clusters from channels. We presented two approaches that allow clustering: The first approach leverages the topical distribution of channels to group actors based on the topical similarity of the content they spread. Applying this to the seed channels for the collection of the dataset indicates promising results for future research attempts in clustering actors on social media based on the content of their postings in a time-saving manner. The second method we propose in this context leverages embeddings learned from the social graph that we generated from the dataset in an unsupervised manner. The advantage of our approach over traditional community detection methods, such as the Louvian method (Blondel et al. 2008), is that it can handle node attributes, meaning additional data can be incorporated in the community detection. This enabled us to combine network data (i.e. relations between the channels) with data about the topics that are discussed in the channels. Our results indicate different communities that vary by the number of hateful users present. Large communities appear to be spanned by seed users which was to be expected based on our data collection approach; however, we also detected smaller communities that do not contain any seed users, indicating that our sampling approach was able to find communities beyond the direct sphere of influence of the initial seed set. For a more precise evaluation of these results, more general information about the German hater community and its relative extend would have been helpful. However, no such studies are currently available.

Conclusion and Future Work

To the best of our knowledge, we are the first to develop abusive language classification models for German messages on Telegram. Our results look promising. The text model outperforms Google's Perspective API in terms of F1 score (macro F1: 73.2%). Similarly, the channel classification model provides good performance in detecting *hater* channels (macro F1: 69.5%). In addition, we have outlined methods for facilitating and scaling abusive language analysis on a message level as well as on the channel level. In the latter case, we fully relied on unsupervised learning methods, which makes these approaches particularly appealing. Furthermore, we publish the first abusive language dataset consisting of German Telegram messages.

There are multiple possible directions for future work in this research field. Firstly, the research community would benefit from larger annotated corpora, which should also include media files shared in those channels (e.g., photos with messages, memes, and videos). Because such media files (e.g., memes) can be used to transport hate (Kiela et al. 2021), they are relevant for the problem of detecting abusive content but were not part of this study.

Regarding the classification model for *hater* channels, integrating additional data (e.g., metadata of the channels) and enhancing the NN architecture could improve classification performance. An explorative network analysis of the subnetwork could help identify additional features and give a better overview of the relative extent of hateful communities on Telegram. In addition, a larger overall sample size of Telegram should be collected to mitigate the selection bias introduced by our selection of hateful seed users.

We also encourage researchers from various core disciplines, such as machine learning and social sciences, to synergize in their research efforts and validate the performances achieved by sophisticated learning frameworks applied to large amounts of data with perspectives from social and political science on these phenomena. Due to the unstoppable increase in content produced on social platforms such as Telegram, automatic methods for generating insights will become indispensable. Finally, the hate speech detection community should look into applying approaches such as the ones presented here to other alternative social media platforms as hate actors will congregate there as deplatforming efforts continue.

References

Angelov, D. 2020. Top2vec: Distributed representations of topics. *arXiv preprint arXiv:2008.09470* .

Baumgartner, J.; Zannettou, S.; Squire, M.; and Blackburn, J. 2020. The Pushshift Telegram Dataset. In *Proceedings of the International AAAI Conference on Web and Social Media*, volume 14, 840–847.

Blondel, V. D.; Guillaume, J.-L.; Lambiotte, R.; and Lefebvre, E. 2008. Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment* 2008(10): P10008. doi:10.1088/1742-5468/2008/10/p10008. URL https://doi.org/10.1088/1742-5468/2008/10/p10008.

Bretschneider, U.; and Peters, R. 2017. Detecting offensive statements towards foreigners in social media. In *Proceedings of the 50th Hawaii International Conference on System Sciences*.

Chan, B.; Schweter, S.; and Möller, T. 2020. German's Next Language Model. In *Proceedings of the 28th International Conference on Computational Linguistics*, 6788–6796. Barcelona, Spain (Online): International Committee on Computational Linguistics.

CSIRO's Data61. 2018. StellarGraph Machine Learning Library. https://github.com/stellargraph/stellargraph. Accessed: 2022-03-31.

Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, 4171–4186. Minneapolis, Minnesota: Association for Computational Linguistics.

Dietterich, T. G. 1998. Approximate Statistical Tests for Comparing Supervised Classification Learning Algorithms. *Neural Computation* 10(7): 1895–1923. ISSN 0899-7667. doi:10.1162/089976698300017197. URL https://doi.org/10.1162/089976698300017197.

Duggan, M. 2017. *Online harassment 2017*. Pew Research Center.

Echikson, W.; and Knodt, O. 2018. Germanys NetzDG: A key test for combatting online hate. *CEPS Policy Insight* .

Eckert, S.; Leipertz, S.; and Schmidt, C. 2021. Querdenker: Wie die Corona-Krise zu Radikalisierung führte. *Norddeutscher Rundfunk* URL https://story.ndr.de/querdenker/. Visited on 11/20/2021.

Fielitz, M.; and Schwarz, K. 2020. *Hate not Found?! Deplatforming the Far-Right and its Consequences*. Institut für Demokratie und Zivilgesellschaft: Jena.

Grave, E.; Bojanowski, P.; Gupta, P.; Joulin, A.; and Mikolov, T. 2018. Learning Word Vectors for 157 Languages. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC 2018)*.

Hamilton, W. L.; Ying, R.; and Leskovec, J. 2017. Inductive representation learning on large graphs. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, 1025–1035.

Hohlfeld, R.; Bauerfeind, F.; Braglia, I.; Butt, A.; Dietz, A.-L.; Drexel, D.; Fedlmeier, J.; Fischer, L.; Gandl, V.; Glaser, F.; Haberzettel, E.; Helling, T.; Ksbauer, I.; Kast, M.; Krieger, A.; Lchner, A.; Malkanova, A.; Raab, M.-K.; Rech, A.; and Weymar, P. 2021. Communicating COVID-19 against the backdrop of conspiracy ideologies: How public figures discuss the matter on Facebook and Telegram .

Holzer, B. 2021. Zwischen Protest und Parodie : Strukturen der Querdenken-Kommunikation auf Telegram (und anderswo). In Reichardt, S., ed., *Die Misstrauensgemeinschaft der Querdenker : Die Corona-Proteste aus kultur- und sozialwissenschaftlicher Perspektive*, 125–157. Frankfurt: Campus Verlag.

Honnibal, M.; Montani, I.; Van Landeghem, S.; and Boyd, A. 2020. spaCy: Industrial-strength Natural Language Processing in Python. URL https://doi.org/10.5281/zenodo.1212303.

Jigsaw. 2021. Perspective API - Case Studies URL https://www.perspectiveapi.com/case-studies/. Visited on 11/20/2021.

Kiela, D.; Firooz, H.; Mohan, A.; Goswami, V.; Singh, A.; Fitzpatrick, C. A.; Bull, P.; Lipstein, G.; Nelli, T.; Zhu, R.; et al. 2021. The Hateful Memes Challenge: Competition Report. In *NeurIPS 2020 Competition and Demonstration Track*, 344–360. PMLR.

Kili Technology. 2021. Text annotation tool. URL https://kili-technology.com. Visited on 11/20/2021.

Krippendorff, K. 2004. *Content Analysis: An Introduction to Its Methodology*. Content Analysis: An Introduction to Its Methodology. Sage.

Kurrek, J.; Saleem, H. M.; and Ruths, D. 2020. Towards a Comprehensive Taxonomy and Large-Scale Annotated Corpus for Online Slur Usage. In *Proceedings of the Fourth Workshop on Online Abuse and Harms*, 138–149. Online: Association for Computational Linguistics.

Li, S.; Zaidi, N. A.; Liu, Q.; and Li, G. 2021. Neighbours and Kinsmen: Hateful Users Detection with Graph Neural Network. In Karlapalem, K.; Cheng, H.; Ramakrishnan, N.; Agrawal, R. K.; Reddy, P. K.; Srivastava, J.; and Chakraborty, T., eds., *Advances in Knowledge Discovery and Data Mining*, 434–446. Cham: Springer International Publishing.

Mandl, T.; Modha, S.; Kumar M, A.; and Chakravarthi, B. R. 2020. Overview of the HASOC Track at FIRE 2020: Hate Speech and Offensive Language Identification in Tamil, Malayalam, Hindi, English and German. In *Forum for Information Retrieval Evaluation*, FIRE 2020, 29–32. New York, NY, USA: Association for Computing Machinery.

Mandl, T.; Modha, S.; Majumder, P.; Patel, D.; Dave, M.; Mandlia, C.; and Patel, A. 2019. 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, 1417. New York, NY, USA: Association for Computing Machinery. ISBN 9781450377508.

Mosca, E.; Wich, M.; and Groh, G. 2021. Understanding and interpreting the impact of user context in hate speech detection. In *Proceedings of the Ninth International Workshop on Natural Language Processing for Social Media*, 91–102.

Müller, K.; and Schwarz, C. 2021. Fanning the flames of hate: Social media and hate crime. *Journal of the European Economic Association* 19(4): 2131–2167.

Rafael, Simone; Ritzmann, A. 2019. *Hate Speech and Radicalisation Online - The OCCI Research Report*, chapter

Background: the ABC of hate speech, extremism and the NetzDG. ISD Global.

Räther, S. 2021. *Investigating Techniques for Learning with Limited Labeled Data for Hate Speech Classification*. Master's thesis, Technical University of Munich. Advised and supervised by Maximilian Wich and Georg Groh.

Ribeiro, M.; Calais, P.; Santos, Y.; Almeida, V.; and Meira Jr, W. 2018. Characterizing and Detecting Hateful Users on Twitter . In *Proceedings of the Twelfth International AAAI Conference on Web and Social Media (ICWSM 2018)*.

Rogers, R. 2020. Deplatforming: Following extreme Internet celebrities to Telegram and alternative social media. *European Journal of Communication* 35(3): 213–229.

Ross, B.; Rist, M.; Carbonell, G.; Cabrera, B.; Kurowsky, N.; and Wojatzki, M. 2016. Measuring the Reliability of Hate Speech Annotations: The Case of the European Refugee Crisis. In Beißwenger, M.; Wojatzki, M.; and Zesch, T., eds., *Proceedings of NLP4CMC III: 3rd Workshop on Natural Language Processing for Computer-Mediated Communication*, volume 17 of *Bochumer Linguistische Arbeitsberichte*, 6–9. Bochum.

Sap, M.; Card, D.; Gabriel, S.; Choi, Y.; and Smith, A. N. 2019. The risk of racial bias in hate speech detection. In *ACL*.

Sap, M.; Swayamdipta, S.; Vianna, L.; Zhou, X.; Choi, Y.; and Smith, N. A. 2021. Annotators with attitudes: How annotator beliefs and identities bias toxic language detection. *arXiv preprint arXiv:2111.07997* .

Sen, I.; Flöck, F.; Weller, K.; Weiß, B.; and Wagner, C. 2021. Applying a total error framework for digital traces to social media research. In *Handbook of Computational Social Science, Volume 2*, 127–139. Routledge.

Solopova, V.; Scheffler, T.; and Popa-Wyatt, M. 2021. A Telegram corpus for hate speech, offensive language, and online harm. *Journal of Open Humanities Data* 7.

Struß, J. M.; Siegel, M.; Ruppenhofer, J.; Wiegand, M.; and Klenner, M. 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)*, 354–365.

Swamy, S. D.; Jamatia, A.; and Gambäck, B. 2019. Studying Generalisability across Abusive Language Detection Datasets. In *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, 940–950. Hong Kong, China: Association for Computational Linguistics.

Urman, A.; and Katz, S. 2020. What they do in the shadows: examining the far-right networks on Telegram. *Information, Communication & Society* 0(0): 1–20.

Wich, M.; Breitinger, M.; Strathern, W.; Naimarevic, M.; Groh, G.; and Pfeffer, J. 2021a. Are your Friends also Haters? Identification of Hater Networks on Social Media: Data Paper. In *Companion Proceedings of the Web Conference 2021 (WWW'21 Companion)*.

Wich, M.; Mosca, E.; Gorniak, A.; Hingerl, J.; and Groh, G. 2021b. Explainable Abusive Language Classification Leveraging User and Network Data. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, 481–496. Springer.

Wich, M.; Räther, S.; and Groh, G. 2021. German Abusive Language Dataset with Focus on COVID-19. In *Proceedings of the 17th Conference on Natural Language Processing (KONVENS 2021)*.

Wiegand, M.; Siegel, M.; and Ruppenhofer, J. 2018. Overview of the germeval 2018 shared task on the identification of offensive language. In *Proceedings of the 14th Conference on Natural Language Processing (KONVENS 2018)*.

Williams, M. L.; Burnap, P.; Javed, A.; Liu, H.; and Ozalp, S. 2020. Hate in the Machine: Anti-Black and Anti-Muslim Social Media Posts as Predictors of Offline Racially and Religiously Aggravated Crime. *The British Journal of Criminology* 60(1): 93–117.