SN Computer Science (2021) 2:95
https://doi.org/10.1007/s42979-021-00457-3

ORIGINAL RESEARCH

Challenges of Hate Speech Detection in Social Media

Data Scarcity, and Leveraging External Resources

György Kovács[1] · Pedro Alonso[1] · Rajkumar Saini[1]

Received: 4 May 2020 / Accepted: 8 January 2021 / Published online: 13 February 2021

Abstract
The detection of hate speech in social media is a crucial task. The uncontrolled spread of hate has the potential to gravely damage our society, and severely harm marginalized people or groups. A major arena for spreading hate speech online is social media. This significantly contributes to the difficulty of automatic detection, as social media posts include paralinguistic signals (e.g. emoticons, and hashtags), and their linguistic content contains plenty of poorly written text. Another difficulty is presented by the context-dependent nature of the task, and the lack of consensus on what constitutes as hate speech, which makes the task difficult even for humans. This makes the task of creating large labeled corpora difficult, and resource consuming. The problem posed by ungrammatical text has been largely mitigated by the recent emergence of deep neural network (DNN) architectures that have the capacity to efficiently learn various features. For this reason, we proposed a deep natural language processing (NLP) model—combining convolutional and recurrent layers—for the automatic detection of hate speech in social media data. We have applied our model on the HASOC2019 corpus, and attained a macro F1 score of 0.63 in hate speech detection on the test set of HASOC. The capacity of DNNs for efficient learning, however, also means an increased risk of overfitting. Particularly, with limited training data available (as was the case for HASOC). For this reason, we investigated different methods for expanding resources used. We have explored various opportunities, such as leveraging unlabeled data, similarly labeled corpora, as well as the use of novel models. Our results showed that by doing so, it was possible to significantly increase the classification score attained.

Keywords Hate speech · Deep language processing · Transfer learning · BERT · Vocabulary augmentation

Introduction

The debate around the regulation of hate speech is still ongoing [9, 25, 34, 51]. It is still not clear whether the best response to it is through legal measures, or other methods (such as counter-speech and education [13]). Regardless of the means of countering it, the evident harm of hate speech [29, 58, 72] makes its detection crucial. Both the volume of content generated online, particularly in social media, and the psychological burden of manual moderation [33] supports the need for the automatic detection of offensive and hateful content.

Related Work

The Internet enables the access and sharing information at an unprecedented rate. This potential combined with the opportunity to remain anonymous [102] also makes it an effective vehicle for the spread of hateful or offensive content. Because of this, many researchers have examined the task of automatically detecting this phenomenon [48, 98], and there were also many competitions dedicated to solving problems like that (e.g. detecting insults [37], aggression [43], or hateful/offensive content in German [89], Spanish [10], or English [49, 95]). A basic approach is using simple templates (e.g. I [intensity] [userintent] [hate target]) [57] or keywords [48] (advocated for by Gröndahl et al. [32]). A

This article is part of the topical collection "Social Media Analytics and its Evaluation" guest edited by Thomas Mandl, Sandip Modha and Prasenjit Majumder.

✉ György Kovács
gyorgy.kovacs@ltu.se

[1] Luleå University of Technology, Aurorum 1, 971 87 Luleå, Sweden

more popular approach is the combination of feature extraction and classical machine learning algorithms. Kwok and Wang [44], used the Bag-of-Words (BoW) approach combined with a Naïve Bayes classifier. While Grevy et al. [31] applied BoW in conjunction with Support Vector Machines (SVMs). But due to the high false positive rates of BoW [19], others applied more sophisticated methods to provide features for classical machine learning methods (e.g. SVM, Naïve Bayes and Logistic Regression) [14, 19, 85, 87, 101]. Salminen et al. [73] experimented with more classes (e.g. accusation, humiliation, etc.) along with hate/non-hate classification. SVM with linear kernel was suggested by them.

After its success in computer vision, pattern recognition, and speech processing, Deep Learning has also gained significant traction in NLP as well during the last decade [92], propelling it forward [64]. One important step in this direction was the introduction of embeddings [56], which proved useful combination with classical machine learning algorithms for hate speech detection [54, 93], highly outperforming the BoW approach [23]. Other Deep Learning methods in the subject include the use of Convolutional Neural Networks (CNNs) [6, 26, 65, 103], Recurrent Neural Networks (RNNs) [6, 20, 24, 84], and a combination of the two [36, 88, 99]. Another important step was the introduction of transformers, particularly BERT [21], which in a recent competition for hate speech detection provided seven out of the ten best performing models in a subtask [95]. It is also possible to use an ensemble of the above methods [59, 61]. In fact, such an approach has recently provided the best performance (based on the average performance on all subtasks) in a competition among more than fifty participating teams [77].

Challenges of Detecting Hateful and Offensive Speech

There are many layers to the difficulty of automatically detecting hateful and/or offensive speech, particularly in social media. Some of these difficulties being closely related to the shortcomings of keyword-based approaches. For one, words can be obfuscated in many different ways, both in an intentional attempt to avoid automatic content moderation [60], or as a consequence of the use of social media for communication (consider, for example the tendency in some posts to replace letters with similar looking numbers, e.g. “E”s with 3s, or “l”s with 1s, and so on).

Furthermore, there are many expressions that are not inherently offensive, however they can be so in the right context [70, 81]. But even in the case of slurs, not only different slurs hold a different degree of offense [69], the offense can also vary based on different time (as previously innocuous words may become slurs in time), as well as different use of the same word, different users, and different audience members [35]. One example of this is the difference in the use of slurs by in-group speakers, and out-group speakers [11]. This factor, when disregarded can contribute to the bias in hate speech detection corpora (in particular against African Americans [18], and more specifically, against African American men [41]), and in turn, the bias in hate speech detection [75] (a strong argument for the transparency and explainability of hate speech detection models).

One recommendation to mitigate bias is explicitly preparing annotators for it [75]. This leads to another difficulty, namely the availability (or lack thereof) of reliably annotated data. A factor that contributes to this problem is that there is no universally accepted definition of hate speech (a statement many publications would agree on [1, 2, 28, 38, 62, 63, 80, 81]), let alone one that is productive. One can point at a United Nations report for definition [5], we would however argue that it does not satisfy the criteria of being a universally accepted productive definition on several accounts. For one, the recommendations in said document are not legally binding [82], thus their implementation in all member countries is not a given. Furthermore, the recommendation here is only to “draw [...] from the guidance and definitions” [5], not to apply them as it is, thus even if the recommendations were binding, or all member countries would decide by their own volition to accept them, different countries could still arrive at different definitions implemented in their domestic legal frameworks. Moreover, even if the definitions were used “as is”, the question still remains whether they would be applicable for large scale data annotation, considering the contextual nature of their terms (“First, one should realize that the question of distinguishing those forms of expression that should be defined as incitement to hatred and thus prohibited is contextual and the individual circumstances and the individual circumstances of each case, such as local conditions, history, cultural and political tensions, must be taken into account” [5]), and the complexity of definitions that could necessitate annotators having a background in law.

One benefit of a universally agreed upon productive definition for hate speech could be important for more reliable annotation, with higher inter-annotator agreement [71]. For example the 2019 HASOC hate speech and offensive content evaluation task [50] had an interrater agreement rate that is between 69 and 77 percent for different task, despite the fact that “many texts recommend 80% agreement as the **minimum acceptable** interrater agreement” [52]. According to Mandl et al. [50] (the organizers of the 2019 HASOC hate speech and offensive content evaluation task), one difficulty in annotation (an issue that may have contributed to the low interrater agreements) was the use of language registers, such as youth talk.

The difficulty of annotating youth talk is exemplified by the annotation of some example tweets (see Table 1) where the name of India’s prime minister (Narendra Modi,

Table 1 Tweets where the name of the Indian prime minister is used in pop-cultural references, and their annotations

Tweet	Annotation
Modi Ji will never give you up Modi ji will never give you down	Not hateful/offensive
Modi Ji knows Coca Cola's secret ingredient	Not hateful/offensive
Modi Ji knows why is Gamora	Hateful/offensive
Modi Ji knows who let the dogs out	Hateful/offensive

or Modi Ji) was used in various pop-cultural references (or ”memes”). The first being a paraphrase of the chorus (*“Never gonna give you up // Never gonna let you down // Never gonna run around and desert you...”*) from the 1987 Rick Astley hit, *Never Gonna Give You Up* (that gained a considerable reputation in recent years, due to its use in the phenomenon called ”rickrolling”). The second being a reference to a popular beverage that is well known for the company's secrecy regarding its recipe. While the third referencing a much quoted part of a recent movie, and the last one making a reference to a 1998 song from Baha Man (*Who let the dogs out*). Despite the similar nature of the tweets (particularly the last three tweets, as all three of them allude to Modi Ji knowing something that in general considered impenetrable - as mentioned before, the ingredients of Coca Cola are considered a well-kept secret; part of the comedic effect of Drax the destroyer asking the question “Why is Gamorra” is derived from the fact that this question itself is considered unanswerable; and not only the song does not answer the question, who let the dogs out, but two of the artists contributing to the song also refused to do so in recent interviews[1]), however, two were labelled as hateful or offensive, while the other two were not. Here, it is important to note that our argument is not that all of these tweets should be annotated as hate speech, but rather that these tweets should have a uniform annotation. And in our opinion, given the innocuous nature of the references, they would be annotated as not hateful, given an annotator who is aware of the cultural context.

Another example that may result from the annotation of hateful or offensive speech being subjective is that of the labeling of tweets containing the word fuck (subsequently referred to as the “f-word”). Particularly, the difference in labeling between the case when the word is used as part of a hashtag, as opposed to when it is used outside of a hashtag. To showcase this, we have examined the first subtask of the English task in the HASOC 2019 competition. In the training set of corpus [49] there are 1159 tweets that contain the “f-word” within a hashtag, and 215 tweets that contain it outside of a hashtag. After eliminating those tweets that contain both forms, these numbers decrease to 1072 and 128, respectively. We find that these two categories of tweets are annotated very differently. When the f-word is used outside of a hashtag, alone in a tweet, more than 97% of the tweets were considered hateful or offensive. However, for the version that is used within a hashtag, this number is only approximately 41% (this value for tweets that do not contain the word at all is approximately 38%). This indicates that while the use of the “f-word” in and of itself highly increases the probability of a tweet deemed as hateful or offensive, tweets that contain it in a hashtag are only slightly more likely to be labeled hateful or offensive than any other tweet.

Another challenge to consider is that of imbalanced data. While the spread of hateful and offensive content is a serious problem in social media, it is fortunately still true that this content only constitutes a small fraction of all content. Part of this imbalance manifests in hate speech corpora as well [98]. In the HASOC 2019 training set for example, less than forty percent of tweets was categorized as hateful or offensive [49]. There are, however, standard methods available to tackle this issue [15], and thus we would not discuss it any further.

Contribution

In the previous section we have outlined some of the major challenges of automatic detection of hate speech and offensive content. In this paper, we focus on the challenge that is posed by the limitation of available data. We do so by extending our previous work [66] on the first English subtask of the 2019 HASOC task. As the training data provided for this challenge was limited in less than six thousand tweets, we contend that it provides an ideal pilot for the examination of data scarcity, and the effect of leveraging external resources. One could argue that the benefits of more data are already well established, but that is not always the case [27, 100]. Furthermore, what we consider an important contribution here, is the systematic examination of leveraging additional data.

We consider three external resources to leverage. First, as many competitions have been organized that are similar to HASOC [10, 37, 43, 49, 89, 95], and many researchers compiled their own corpora as part of their studies [17, 19], we sought out similarly labeled, self-contained corpora. Second, we considered pretrained word representations [30]. Lastly, we also considered models pretrained on more generic NLP tasks.

[1] https://www.huffpost.com/entry/who-let-the-dogs-out-no-really-who-did-it_n_578d336ae4b0a0ae97c2fee4?section=weird-news.

Table 2 Samples from the English language training data of HASOC 2019 along with their ground truth labels (for a more detailed explanation of the labels used in this sub-task, please read Sect. 2.1.1)

Post	Ground Truth
@TajinderBagga @fawadchaudhry 2uat @cricketworldcup is doing they have objection on #DhoniKeepsTheGlove	NOT
@brooklynmarie @PressSec @realDonaldTrump Amen to that! #fucktrump	NOT
#FuckTrump ANOTHER ONERY DEM..... https://t.co/vXVoZRATpX	HOF
@republic Then shouldn't all who made memes n gave gaalis to PM Modi, also apologise?!	HOF

Delimitations

Although we consider our paper to be beneficial for all who are interested in the subject of hate speech detection, to best understand its context, it is important to note that our work reported here is an extension of the work Alonso et al. carried out for the HASOC 2019 competition [66]. One consequence of this fact is that our main focus remained on the very same dataset. It would thus be the subject of a further study to examine whether our findings reported here generalize to other hate speech detection datasets too.

Furthermore, as discussed in Sect. 1.3, our focus here was on one particular problem, that is the challenge posed by the limitations of available labeled data. This means that our experiments and suggestions address this problem. This also means that our discussion of other problems regarding hate speech detection are limited, and the purpose of this discussion is mostly to provide proper context. This also means that further comments on efforts towards introducing universally accepted hate speech definitions, handling bias, or dealing with unbalanced data (particularly in terms of the difficulty it introduces to data collection and annotation [68]) are outside of the scope of this paper.

Lastly, we approach the problem of hate speech detection from the perspective of deep learning. This decision was motivated by the recent success of deep learning in various text classification[2] [78] and sentiment analysis[3] tasks. Thus the methods we examine here will be deep learning methods that learn their representation not from manually selected features, but from raw text data by statistical means. This means that we also did not carry out pos tagging, stemming, or lemmatisation.

Structure

Following the introduction, the discussion regarding the subject of hate speech detection in this paper will be as followed. First, in Sect. 2 we discuss the external data sources to leverage in more detail. Then, in Sect. 3 we describe the different machine learning methods we applied on the resulting data. After which, we present and discuss our experimental results in Sect. 4, then close this study with conclusions and future work in Sect. 5.

Materials

In this section we discuss in detail the challenge (that is automatic hate speech detection on the HASOC 2019 dataset [50]) we undertake (including the task of the challenge and the database), and the resources outside of the challenge that we examined.

Labeled Corpora

As we consider the automatic detection of hateful or offensive speech as a supervised classification task, an important resource to discuss is labeled corpora. Below, we detail three such corpora that we used in our experiments, including the corpus used in the HASOC 2019 challenge we primarily target to solve. Here, for the sake of repeatability, as tweets (especially those that are offensive in nature) may be erased from twitter [68] we only considered self-contained datasets, where tweets are also made available for download (as opposed to only ids being uploaded). This means that some popular datasets had to be excluded from our examination [86, 87].

HASOC 2019

What we consider our main challenge in this paper is the first sub-task of the "Hate Speech and Offensive Content Identification in Indo-European Languages" challenge [50] (HASOC 2019). For this task a corpus[4]—that will serve as our primary resource—was created consisting of social media posts in three languages (English, German, Hindi) from Twitter and Facebook. The English language data consisted of 7551 instances: 6358 (approximately 85 percent), for training, and 1153 for testing purposes. Some examples

[2] https://paperswithcode.com/task/text-classification.

[3] https://paperswithcode.com/task/sentiment-analysis.

[4] Available for download at: https://hasocfire.github.io/hasoc/2019/dataset.html.

Table 3 Samples from the HateBase corpus along with their ground truth labels

Post	Annotators				Ground
	No.	Hate	Offensive	Neither	Truth
		Speech	Language		Label
"Subtweet me one more time, you dirty chink whore"	3	3	0	0	Hate Speech
Switcharoo on yo main bitch me and my NIGGGA hell naw we don't date hoes	6	1	5	0	Offensive Language
Stayin up late to watch #Ichiro #Yankees	3	0	0	3	Neither
Taco Bell is super trash when you're sober	3	0	1	2	Neither

of the training set (along with their ground truth labels) are shown in Table 2.

The HASOC 2019 task description [49, 50] defines the first sub-task in the hate speech detection challenge as follows: The task here is a more general binary classification of social media posts into two categories, namely the "Hate and Offensive" category (HOF) and the "Non- Hate and offensive" category (NOT). Posts here are annotated as follows:

- NOT: These are posts without sentences considered to be hate speech or offensive in content.
- HOF: These posts are considered to contain hateful, offensive or profane language.

OLID

The first external data source we examined was the OLID[5] corpus (discussed in more detail in [94]), that has been used in one of the 2019 SemEval tasks [95]. We did so motivated by the similarity of the two corpora. As both corpora contained mainly English language tweets, and were labeled similarly, applying a three-level hierarchical annotation schema, with the first level being a decision whether a tweet is offensive or not. One also finds very close similarity when looking at the definition of the two labels in the first level of annotation. In the case of OLID, posts that are deemed not offensive are *"Posts that do not contain offense or profanity"*, while posts labeled as offensive are "Posts containing any form of non-acceptable language (profanity) or a targeted offense, which can be veiled or direct. This includes insults, threats, and posts containing profane language or swear words" [94]. Here, the corpus contains 13240 tweets, 4400 (approximately 33 percent) labeled as hateful or offensive (Table 3).

The use of labeled data from other sources can be particularly interesting, given that the generalization capability of models for the task has been quite low even in the case of similar datasets [4]. Thus it was unclear to what degree similarly labeled datasets, collected and annotated at different times by different people can be helpful in classification of HASOC data.

HateBase

Davidson et al. also created their corpus when working on the automatic detection of hateful and offensive content [19], and made it available on Github [17]. The labeled data we downloaded contained 24783 tweets annotated into three classes, namely hate speech (1430 tweets), offensive language (19190 tweets), and neither (4163). Here, to limit the use of additional data, we only used tweets from the first and third category. An advantage of this database was that not only the final labels are available, but one can also learn how many annotators labeled a given tweet, and how many votes each label got before the final decision. Thus one could also limit training to those instances where all annotators agreed.

Pretrained Models

One popular method for leveraging further data sources is that of transfer learning, particularly the case where a model trained on a more general task, for which there is an abundance of data available, is then fine-tuned on a specific task with limited training data. Here, we also applied this method using a BERT variant, namely RoBERTA [47], that we discuss in more detail in Sect. 3.3.2.

Word Embeddings

Another option to leverage large amounts of unlabeled data is through the use of pre-trained word representations. A prominent example is Word embeddings, that are mappings of words from natural language to a real vector space (e.g. FastText [12], GloVE [67]). These mappings have become crucial in NLP for a variety of tasks, including machine translation, and sentiment analysis. In our experiments, we used wiki-news-300d-1M, a 300 dimensional word

[5] Available for download at: https://scholar.harvard.edu/malmasi/olid.

embedding of 1 million word vectors trained on Wikipedia 2017, UMBC webbase corpus and the statmt.org news dataset [55].

Methods

In this section we discuss our methodology. First, the text preprocessing methods applied are described in Sect. 3.1. Then, we discuss the cross-validation method used to train and evaluate our models (Sect. 3.2). Lastly, we discuss the models we trained in more detail in Sect. 3.3.

Text Preprocessing

The mentioned datasets/corpora were preprocessed before they were fed into the model. The preprocessing involves replacements/removal of certain tokens types. Each tweet was tokenized into words. Extra spaces were removed. The $@-words$, and URLs were replaced with $@USER$, URL respectively. Hash characters (#) and emoticons were also removed from the tweets.

We investigated the contribution of emojis to hatefulness. To measure this, we calculated the Pearson correlation coefficient [42] between scores of hatefulness/offensiveness and emojis on the training set published for the OffensEval 2020 competition [96]. For this, we checked the correlation coefficient between the emoji-count (number of emojis in a tweet) and the scores. We found out that emojis were not really correlated with tweet scores based on the 9 million tweets available in OffensEval, as the correlation was very close to zero (−0.046). Similarly, the correlation between the facial emotion emojis and the tweet scores was also investigated. It was found that the facial emotion emojis and the tweet scores were also not correlated.

Based on the results of our experiments on correlation, all emojis were removed from the tweets in our preprocessing step. We also removed emoticons from the tweets. We, however, did not remove any other special character. Furthermore, no spelling correction was done. We also did not use stemming or lemmatization. We followed minimal preprocessing to investigate the effectiveness of the proposed model. In the end, the processed tweets were generated with only one space character between the words/tokens in each tweet. Thus, the input to the network were the tweets after the above-mentioned preprocessing steps. The whole preprocessing was done with the NLP toolkit of MATLAB.[6]

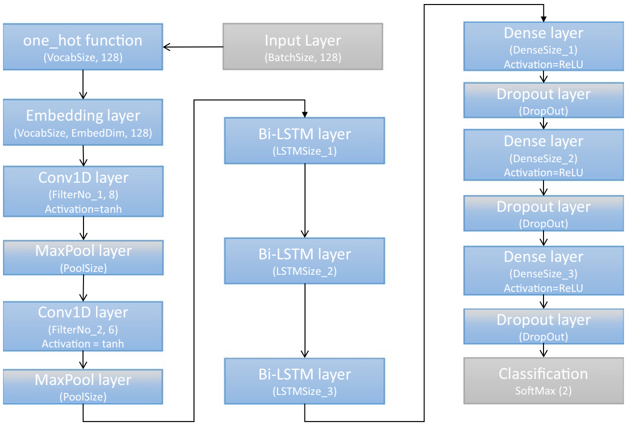


Fig. 1 Architecture of the CNN-BiLSTM model used in our experiments

Cross-Validation

Here, training for all the machine learning methods we utilized was carried out using a form of fivefold cross-validation, as follows. First, we partitioned the training set of the HASOC data into five development sets without overlap, in a manner that preserved the class distribution of the training set as closely as possible. In each case, the remaining training data (data not included in the development set) was used as the training set. Then, for each machine learning method (that we applied here), we trained five separate models, each using a training set for parameter-optimization, and a development set for validation purposes. The final decision for each method was then attained by calculating the predicted probabilities using all five models, then averaging these probabilities, and lastly classifying each instance using the label with the higher predicted probability. When using further corpora, we added the new training data to all five training folds.

Deep Learning Methods

Given the success of deep learning approaches for text classification in general [78], in the task of hate speech detection [6, 103], and more particularly, in recent competitions in the subject [50, 84, 95][7], we decided to focus in our experiments on deep learning methods. First, similar to Do et al. [36], we used a combination of Convolutional and Long Short-Term Memory (LSTM) layers (CNN-LSTM), then we also conducted experiments with a popular transformer model (RoBERTa). Following this, we also experimented

[6] https://in.mathworks.com/discovery/natural-language-processing.html

[7] In the 6th task of the 2019 SemEval competition, 70% of the contributions (including those of the top 3 teams) applied a deep learning method

Table 4 Architectural parameters

Parameter name	Parameter value
vocabulary_size	5000, 10000, 20000, 30000, 40000, 50000
batch size	32, 64
CNN layer 1	64, 128
CNN layer 2	0, 32, 64
Pooling size	2, 4
LSTM layer 1	600, 800, 1000
LSTM layer 2	0, 400, 600, 800
LSTM layer 3	0, 100, 200, 400
Dense layer 1	200, 400, 600
Dense layer 2	0, 100, 200, 400
Dense layer 3	0, 100, 200
dropout	0.2, 0.3, 0.4, 0.5
embedding dimension	100, 150, 200, 250, 300, 350, 400
learning_rate	0.0005, 0.001, 0.002

with RoBERTa as a feature extractor for classical machine learning methods. Lastly, we also carried out experiments using the FastText classification model. These models are discussed below in more detail.

CNN-LSTM

In their contribution to the 2019 HASOC challenge, Alonso et al. [66] introduced a simple CNN-LSTM architecture. For this study we have used a similar architecture as a basis for hyper-parameter optimisation. As can be seen in Fig. 1, the model begins with an input layer with dimensions defined by the *batch size* parameter, and the maximum tweet length (in terms of tokens) we considered, that in our case is 128. For this layer to be able to handle text input, the Keras *one_hot* function had to be applied, that hashes tokens (with the exception of punctuation) without unicity being guaranteed, to a number, in the range of the *vocabulary size* parameter. The input layer is followed by an embedding layer defined by the *vocabulary size* and the *embedding dimension* parameters, with the input truncated (or padded depending on the number of tokens in the original tweet) to a hundred dimensional vector. Then (similar to the works of [6] and [99]), we used up to two convolutional layers (applying the tanh activation function) with a MaxPooling layer after each. The kernel size of these convolutional layers was 8 and 6 respectively, while the number of filters was another hyper-parameter to be optimized for both layers. After the pooling layer, based on our preliminary experiments, up to three bidirectional LSTM (Bi-LSTM) layers were deployed. The recurrent layers were followed by up to three dense layers each coupled with a dropout layer applying the same *dropout rate* parameter. The last layer was a Softmax layer containing two neurons, corresponding to the two target classes (not hateful or offensive/hateful or offensive).

To optimize these hyper-parameters, for each setting we trained independent networks on the five folds three separate times, and used the average macro F_1-score on the development set to choose the best performing parameter set out of a hundred different, randomly selected parameter sets. The potential values for the various parameters are listed in Table 4.

RoBERTa

Another method we used in our study is one of the many variants of the recently introduced BERT architecture [22], a member of the transformer family. Transformers were first described by Vaswani et al. [83], as a proof that the attention mechanism introduced for recurrent encoder-decoder architectures [7] does not require recurrent cells. To do so, Vaswani et al. in their paper [83] attained an improvement in the task of translation with the use of the attention mechanism without relying on recurrent neural networks, paving he way for further transformer architectures, such as BERT [22]. The success of which lead to the appearance of many variants, including DistilBert [74], AlBERT [45], TinyBERT [39], and RoBERTa [47]. Here, based on our preliminary experiments with DistillBERT and RoBERTa on the OffensEval 2020 hate speech detection task [96], we decided on the use of RoBERTa (that is part of the Simple-Transformers library [90]).

In our experiments we have fine-tuned the RoBERTa model using the cross-validation method described in Sect. 3.2. As training data we used both the HASOC training data only, as well as the combination of OLID data and the HASOC training data. We did so by using the default meta-parameters presented in [90], with the exception of the number of epochs (that was set to maximum 20, but as we also applied early stopping using the development set, we have never reached this limit), and the learning rate, which was equal to 1e-5. Results of these experiments are presented in Sect. 4.

RoBERTa as Feature Extractor for Classical Machine Learning Methods

The features needed for these methods can be extracted from the data manually, but one can also do so automatically. Here, we used representations from a deep architecture for the purpose of extracting the necessary input features from the text data. More specifically, the inputs to the classification layer of the RoBERTa model served as the features in our experiments with classical machine learning methods. As our models, the MATLAB implementation of K-nearest neighbours (K-NN) [16], AdaBoost [76], linear

Table 5 Performance of the CNN-LSTM model (reported scores are the average of five models; the best result in each column is emphasized in bold)

No. of parameters	Databases		F-scores and deviations			
(In millions)	OLID	HASOC	Macro-F_1	STDEV	Weighted-F_1	STDEV
15	-	✓	0.6557	0.0118	0.7217	0.0148
20	✓	✓	**0.7486**	0.0135	**0.8063**	0.0118

discriminant [53], logistic regression [91], random forest [46], and support vector machine [8] were used in this paper. Brief descriptions of these methods are given below.

The K-NN [16] is a method that assigns a class to examples based on the class of their neighbours. K-NN searches over all train examples and finds the k nearest neighbours to a test example. Then, it chooses the most common class among these neighbours, and assigns it to the test example.

Adaptive boosting (AdaBoost) [76] combines the outputs of many weak learners (e.g. decision stumps) through weighting to produce the final weighted output. This method focuses more on the examples that are misclassified by the learners and adjusts the weights in order to improve classification.

Linear discriminant [53] assumes that different classes generate data based on different Gaussian distributions and attempts to distinguish them linearly. It maximizes the ratio of between-class variance and within-class variance. It [97] attempts to find an optimal linear transformation, which can retain the class separablility while reducing the variation within each class.

Simplest logistic regression [91] is a binary classifier that classifies the examples into two classes. It models the probability using the sigmoid function:

$$\frac{1}{1+e^{-\theta^T P}}, \tag{1}$$

parameterized by θ where P represent training examples.

Random forest [46] is an ensemble learning method that uses bagged trees. It fits several decision trees by selecting random examples from a given set of training examples. Each tree votes a class for a test example. Finally, the test example is assigned the class with highest votes.

The two-class Support Vector Machine (SVM) [8] finds the hyperplane associated with the maximum margin of separation of two classes. It treats the classes as positive and negative ($S_i = \{-1, 1\}$). Given a set of training examples $P = \{P_i\}_{i=1,2,...}$ and their labels S_i, it evaluates the following expression (eq. 2) to predict the class ($\hat{S}$) of a test sample T.

$$f(T) = b + \sum_i \alpha_i S_i K(T, P_i), \tag{2}$$

$$\hat{S} = sign(f(T)), \tag{3}$$

where α_i, b are SVM parameters and K is an (in our case linear) SVM kernel.

The results of our experiments with the six different classical machine learning methods are also described in Sect. 4.

FastText

We have also taken use of the FastText text classification models introduced by Joulin et al. [40], using the code made freely available online.[8] When using this method, we carried out model training and evaluation as follows. First, we used the automatic parameter optimization method to train each model. During this process we used the current development set, and ran the parameter optimization in a way that those parameters should be selected that optimize the F_1-score on this development set.

We fixed only two parameters. One was the autotune-duration, which limited the time to 10 minutes. The other was the dimensionality of word embeddings. In case we used the 1 million word vectors trained on the Wikipedia 2017 corpus, the UMBC webbase corpus and statmst.org news dataset that is the Wikinews 300 dimensional word embedding,[9] we set the embedding dimension to be 300. Results of these experiments are also presented in Sect. 4.

Results and Discussion

In this section we present the results of our experiments we attained using different models. As our primary goal is to compare the performance of models in different circumstances, the results for each model are presented in separate subsections. Moreover, for each model we would present our results using the same structure. First, we would present those results where (besides the additional data already inherent to the model) we only used the HASOC data for training. Then, each subsequent result reported is with the use of more and more additional resources. We chose this method of presentation, so as to further emphasize the effect of using additional resources on the resulting classification scores.

[8] https://github.com/facebookresearch/fastText.

[9] Available for download at: https://fasttext.cc/docs/en/english-vectors.html.

Table 6 Performance of the RoBERTa model (the best result in each column is emphasized in bold)

OLID	HASOC	Macro-F_1	Weighted-F_1
–	✓	**0.7945**	**0.8426**
✓	✓	0.7779	0.8197

Lastly, following the practice of the overview paper reporting the results of the HASOC 2019 challenge [50], for each model we present the macro F_1-score, and the weighted F_1-score, and where applicable, the standard deviation of these scores as well.

CNN-LSTM

Here, we report our results obtained with a modified version of the model used by Alonso et al. in the HASOC competition [66]. In this case we carried out the cross-validation five times, thus reported results are the average scores attained by five separate models, which also allowed us to carry out significance tests (a two-tailed heteroscedastic t-test). We should also note that while majority voting attained better scores in many cases, in future experiments we opted for the average of probability predictions, so as not to lose information.

Results attained with the CNN-LSTM model are listed in Table 5. As can be seen in Table 5, the addition of OLID data significantly increased both F-scores (with $p < 0.00002$). It should also be noted, that for the two different settings we ended up with two different models after the hyper-parameter optimization. Furthermore, the model applied in the second case had five million more parameters than the previous one. To make sure that the increase in performance is not entirely due to these factors, we also trained the same model we used for HASOC data only on OLID and HASOC data. The resulting scores were 0.7146 and 0.7784 for macro F_1 -score and weighted F_1-score respectively. An increase compared to the models using HASOC data only, that is in both cases significant again (with $p < 0.0005$).

RoBERTa

We have repeated the above experiments using a pretrained RoBERTa model as well. The results of these experiments are listed in Table 6. As can be seen in Table 6, in this case the addition of further data (beyond the data used in pretraining RoBERTa) did not lead to an increase in classification score. One can also see that the use of RoBERTa markedly increased our results, in fact the F_1-scores attained are better than those reported as the best results in the first sub-task of the HASOC 2019 competition (a macro F_1-score of 0.7882, and a weighted F_1-score of 0.8395 [50]). This means that our results reported here not only compare favorably to those reported in [84], but also to those reported in the 11 other papers whose results are aggregated in [50].

Table 7 Performance of classical machine learning methods when trained on HASOC data (the best result in each column is emphasized in bold)

	Probability summation		Majority voting	
	Macro-F_1	Weighted-F_1	Macro-F_1	Weighted-F_1
K-NN	0.5572	0.6560	0.5674	0.6868
AdaBoost	0.5764	0.6850	0.5622	0.6911
Linear Discriminant	**0.6366**	**0.7231**	**0.6394**	**0.7269**
Logistic Regression	0.5757	0.7125	0.5985	0.7180
Random Forest	0.5571	0.7014	0.5481	0.6974
SVM	0.6011	0.6946	0.5932	0.6877

Table 8 Performance of classical machine learning methods when trained on both HASOC and OLID data (the best result in each column is emphasized in bold)

	Probability summation		Majority voting	
	Macro-F_1	Weighted-F_1	Macro-F_1	Weighted-F_1
K-NN	0.5281	0.6585	0.5213	0.6690
AdaBoost	0.5608	0.7097	0.5596	0.7094
Linear Discriminant	**0.6413**	**0.7448**	**0.6420**	**0.7455**
Logistic Regression	0.5715	0.7109	0.5531	0.7052
Random Forest	0.5244	0.6936	0.5084	0.6840
SVM	0.6268	0.7353	0.6242	0.7323

RoBERTA as Feature Extractor for Classical Machine Learning Methods

The training of the classical machine learning methods applied in our study was carried out following the same 5-fold cross-validation scheme as discussed in Sect. 3.2. Here (as discussed in Sect. 3.4), before being used as features, text data is first fed as input to the pretrained RoBERTa network we downloaded from the Huggingface repository. For each classifier, the performance was measured in two modes, namely by averaging the probability scores predicted by the five different versions and by majority voting.

Table 7 shows the results of the experiments carried out with different classical machine learning methods when only the HASOC data was made available for them. As can be seen in in Table 7, the Linear Discriminant had the best performance, however, even aided by the representative power of RoBERTA, these results are below those we got with our CNN-LSTM model.

Table 9 Performance of the FastText model (reported scores are the average of five models; the best result in each column is emphasized in bold)

Datasets			Word embedding	F-scores and deviations			
HateBase	OLID	HASOC	WikiNews-300D	Macro-F_1	STDEV	Weighted-F_1	STDEV
–	–	✓	–	0.6736	0.0282	0.7462	0.0293
–	–	✓	✓	0.7107	0.0031	0.7825	0.0053
–	✓	✓	–	0.7494	0.0115	0.8102	0.0113
–	✓	✓	✓	**0.7631**	0.0072	0.8245	0.0072
✓	–	✓	–	0.6971	0.0069	0.7828	0.0048
✓	–	✓	✓	0.7064	0.0258	0.7925	0.0125
✓	✓	✓	–	0.7518	0.0044	0.8184	0.0030
✓	✓	✓	✓	0.7591	0.0092	**0.8252**	0.0043

Table 8 shows the results of the experiments carried out with different classical machine learning methods when both OLID and HASOC training data was made available for the models. One can see in Tables 7 and 8 that in both cases linear discriminant outperforms the other methods. It should also be noted that the best overall scores were obtained when both OLID and HASOC data was used. What is more, the SVM classifier trained on the combination of OLID and HASOC data also markedly outperformed its counterpart that only had access to HASOC data. There were, however, also classifiers where this was not the case, and examining this can be a candidate for future work. Furthermore, we can also see that our CNN-LSTM model outperformed the best performing model here, even without the use of additional OLID data.

FastText

Results for our experiments carried out with the FastText classifier are listed in Table 9. As can be seen in Table 9, each model using additional data outperforms the model using only HASOC data. This improvement being significant in almost all cases ($p < 0.05$) with one exception, namely the addition of HateBase data only, in which case although the average F-scores attained were higher for both weighted F_1-score and macro F_1-score, this increase was only significant for the weighted F_1-scores, but not for the macro F_1-scores. This may be due to the limited amount of additional training data (5594 tweets) that we included from the HateBase database. This is a phenomenon that in a future study could be explored in more detail.

One can also see in Table 9 that the improvement in weighted $F1$-score was the smallest with the use of the WikiNews word embedding. Here, an additional difference between the two models was that as the word embeddings were of 300 dimensions, the dimension of vectors had to be fixed during training. This was not the case when the model only used HASOC data, thus it was possible that the lower performance attained without pretrained word embeddings was due to overfitting of the aforementioned parameter. To examine this possibility, we repeated the experiment using only the HASOC training data with a vector size fixed to 300, and attained a classification performance that is lower (a macro F_1-score of 0.6697, and a weighted F_1-score of 0.7389).

Another question we examined was whether the improvement attained by adding the additional training data was due to the addition of labeled data, or due to the increased number of text that can be used for training word representations. For this, we trained a classifier on OLID training data only (using the HASOC development set for validation), and used the resulting word vectors in training new FastText models on the OLID data. Our results showed that while the use of these word representations increased the F-scores attained (we got a macro F_1-score of 0.6973, and a weighted F_1-score of 0.7740), the results attained using both OLID and HASOC labeled data was significantly higher (with $p < 0.001$).

Lastly, as a sanity check we also examined the question whether a significantly better performance could have been achieved had we eliminated punctuation in the preprocessing step. For this we have repeated the experiments in the first two rows on HASOC data after eliminating punctuation from the text. Results of these experiments when using only HASOC data suggest a decreased performance (with a macro F_1-score of 0.6670, and a weighted F_1-score of 0.7363). This difference, however, is not significant. But we also see a similar decrease in performance when using both HASOC data and the WikiNews pretrained word representations (with a macro F_1-score of 0.7045, and a weighted F_1 -score of 0.7694). This difference, however is significant, with $p < 0.05$ in the case of macro F_1-score, and $p < 0.005$ in the case of weighted F_1-score.

Table 9 also shows that the addition of pre-trained word representations lead to an improvement in all cases, regardless of the number of labeled datasets used. The improvement in macro F_1-score, however was only significant (with $p < 0.05$) in one case, namely when using only the OLID dataset. While the improvement in weighted F_1-score was significant (with $p < 0.05$) in three out of four cases, with the exception of the case when both HateBase and HASOC data was used in training.

We should also note here, that the resulting models in the case of FastText were much bigger (the binaries usually took between one and two GigaBytes after training) than those used by the CNN-LSTM (the binaries being smaller than 200 Mbytes), thus we find it promising that by only leveraging some additional training data, the CNN-LSTM model outperformed the FastText model that only used the HASOC data. Moreover, when both models use HASOC and OLID data for training, there is no significant difference between the performance of our CNN-LSTM model, and the FastText model. Given the difference in model size, and the limited time we had for hyper-parameter optimization, this also suggests that with a more extensive hyper-parameter optimization our method could potentially provide better results than those we got using FastText.

Ensemble of Classifiers

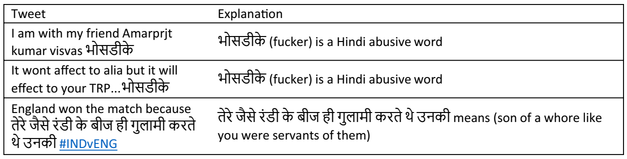
As the overall best performing model in task 6 of the 2019 Semeval competition was an ensemble of different models [77], we also experienced an ensemble solution. For this experiment, to still incorporate additional data in RoBERTa, we combined the RoBERTa model trained on HASOC data with the FastText model trained on all databases, using pretrained word embeddings. We did so by averaging the predicted probabilities provided by the two different models. Using this ensemble, we attained an average macro F_1-score of $\mathbf{0.7953}$, and an average weighted F_1-score of $\mathbf{0.8498}$. Achieving a slight improvement over the RoBERTa model in terms of macro F_1-score, and a more marked improvement in terms of the weighted F_1-score.

Error Analysis

To learn more about the models trained, we listed the instances from the test set that

1. the RoBERTa model misclassified
2. all FastText models misclassified
3. all ensemble models misclassified

Although we did not perform a systematic analysis, we had some interesting findings. One was that our models were seemingly more sensitive to the #F*cktrump hashtag than annotators. As in the test set this occurred 50 times, with 19 occurrences being labeled as not hateful or offensive. The Roberta model however, labeled 15 out of these 19 as offensive, and only labeled 11 of the offensive occurrences as not offensive. For example, the following tweet was classified as hateful or offensive by both RoBERTa, as well as all ensembles: *”#MAGA??? #FUCKTRUMP! #FUCKYOUREPUBLICANS!”*, and our opinion is that here, the automatic method classified this tweet correctly. And the same can be told about the following tweet: *”#America #hypocrites #FuckTrump #FuckICE #FuckYourFlag #FeedTheChristiansToTheLions #TrumpConcentrationCamps #TrumpHasKidsInCages #trumpIsAKiller #TrumpHotels #trumpIsAKiller #LostTrumpHistory”*.



Tweet	Explanation
I am with my friend Amarprjt kumar visvas भोसडीके	भोसडीके (fucker) is a Hindi abusive word
It wont affect to alia but it will effect to your TRP…भोसडीके	भोसडीके (fucker) is a Hindi abusive word
England won the match because तेरे जैसे रंडी के बीज ही गुलामी करते थे उनकी #INDvENG	तेरे जैसे रंडी के बीज ही गुलामी करते थे उनकी means (son of a whore like you were servants of them)

Fig. 2 Hateful or offensive tweets where the hateful/offensive part is in Hindi

Another interesting phenomena was that of mixed language tweets, where the offensive content was seemingly in a foreign language. Figure 2 lists some of the tweets that contain text in Hindi, which we suspect to be labeled offensive solely based on the Hindi text. This may suggest that the task of detecting hateful and offensive text should be considered as a multi-language problem. And even when the primary language of the examined text is English, additional languages should still be considered.

Conclusions and Future Work

In this study we examined the effect of leveraging additional data (both labeled and unlabeled) on the task of automatic detection of hateful and offensive speech through the example of the HASOC 2019 challenge. We did so using several classical machine learning methods, as well as different deep learning models. Our results show that the use of additional labeled data was useful in most cases, even if it was from a different data collection. We have also found, that even in the cases where the additional labeled data failed to increase the recognition score, some form of additional data was helpful. We have also found, however, that not all types of labeled data was useful. It would however take a more thorough examination to draw conclusions regarding what types of labeled data can be successfully leveraged. Moreover, using a combination of RoBERTa and FastText we have attained results that were state-of-the-art at the time of submission. Lastly, it is important to note here, that while in these experiments results attained with RoBERTa were not improved by the use of additional labeled training data, one can find opposing results in the literature. After the initial submission of this paper, Alonso et al. [3] used 1 million tweets from the OffensEval training set to fine-tune the same RoBERTa model, then fine-tuned the resulting model further using the same five folds that we applied in our experiments, and found that the resulting ensemble performed better than the ensemble without fine-tuning. Furthermore, when comparing the performance of the individual models using a paired *t*-test, they found that the additional fine-tuning significantly improved their performance.

Here, we focused on only one dataset, and one problem contributing to the difficulty of the task. In the future we would extend this focus to additional datasets. For example, a model can be pretrained on the larger OffensEval 2020 corpus, and then fine-tuned using HASOC, or a different, smaller dataset. In this study this was not possible due to computational limitations. One may also consider the use of word embeddings that had been trained on twitter data. We also plan to experiment with data augmentation e.g. the introduction of typos or censoring in words, or the use of synonyms in tweets to increase the available training data. Finally, there are many issues that we have not considered here. The one that may be the most important is that of explainability and transparency. For this, in the future we plan to extend our experiments with the use of more explainable models, as well as a more thorough examination of the explainability of our current models (transformer models for example have been successfully examined using the Captum tool [79]).

Acknowledgements Part of this work has been funded by the Vinnova project ”Language models for Swedish authorities” (ref. number: 2019-02996)

Funding Open Access funding provided by Lulea University of Technology.

Compliance with Ethical Standards

Conflict of Interest The authors declare that they have no conflict of interest.

References

1. Alkiviadou N. The legal regulation of hate speech: the international and European frameworks. Politička misao. 2018;55:203–29. https://doi.org/10.20901/pm.55.4.08.
2. Alkiviadou N. Hate speech on social media networks: towards a regulatory framework? Inf Commun Technol Law. 2019;28(1):19–35. https://doi.org/10.1080/13600834.2018.1494417.
3. Alonso P, Saini R, Kovács G. Hate speech detection using transformer ensembles on the hasoc dataset. In: Speech and Computer: 22nd International Conference, SPECOM 2020, St. Petersburg, Russia, October 7–9, 2020, Proceedings, vol. 12335, p. 13. Springer Nature; 2020.
4. Arango A, Pérez J, Poblete B. Hate speech detection is not as easy as you may think: A closer look at model validation. In: Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR'19, p. 45–54. Association for Computing Machinery, New York, NY, USA; 2019. https://doi.org/10.1145/3331184.3331262.
5. Assembly UNG. Annual report of the united nations high commissioner for human rights, report of the united nations high commissioner for human rights on the expert workshops on the prohibition of incitement to national, racial or religious hatred; 2013. https://www.ohchr.org/Documents/Issues/Opinion/SeminarRabat/Rabat_draft_outcome.pdf.
6. Badjatiya P, Gupta S, Gupta M, Varma V. Deep learning for hate speech detection in tweets. In: Proceedings of the 26th International Conference on World Wide Web Companion, WWW '17 Companion, pp. 759–760; 2017.
7. Bahdanau D, Cho K, Bengio Y. Neural machine translation by jointly learning to align and translate. 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. arXiv:1409.0473.
8. Bahlmann C, Haasdonk B, Burkhardt H. Online handwriting recognition with support vector machines-a kernel approach. In: Proceedings Eighth International Workshop on Frontiers in Handwriting Recognition, pp. 49–54. IEEE; 2002.
9. Barendt, E.: What is the harm of hate speech? Ethic theory. Moral Pract. 22, 2019. https://doi.org/10.1007/s10677-019-10002-0.
10. Basile V, Bosco C, Fersini E, Nozza D, Patti V, Rangel Pardo FM, Rosso P, Sanguinetti M. 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, pp. 54–63. Association for Computational Linguistics, Minneapolis, Minnesota, USA; 2019. https://doi.org/10.18653/v1/S19-2007. https://www.aclweb.org/anthology/S19-2007.
11. Bianchi C. Slurs and appropriation: an echoic account. J Pragmat. 2014;66:35–44. https://doi.org/10.1016/j.pragma.2014.02.009.
12. Bojanowski P, Grave E, Joulin A, Mikolov T. Enriching word vectors with subword information. arXiv:1607.04606; 2016.
13. Brown A. What is so special about online (as compared to offline) hate speech? Ethnicities. 2018;18(3):297–326. https://doi.org/10.1177/1468796817709846.
14. Burnap P, Williams ML. Cyber hate speech on twitter: an application of machine classification and statistical modeling for policy and decision making. Policy Internet. 2015;7(2):223–42. https://doi.org/10.1002/poi3.85.
15. Chawla N. Data mining for imbalanced datasets: an overview, vol. 5. New York: Springer; 2005. p. 853–67.
16. Coomans D, Massart DL. Alternative k-nearest neighbour rules in supervised pattern recognition: part 1. k-nearest neighbour classification by using alternative voting rules. Anal Chim Acta. 1982;136:15–27.
17. Davidson T. Hate speech and offensive language. https://github.com/t-davidson/hate-speech-and-offensive-language; 2019. Accessed on 25 Mar 2020.
18. Davidson T, Bhattacharya D, Weber I. Racial bias in hate speech and abusive language detection datasets. In: Proceedings of the

Third Workshop on Abusive Language Online, pp. 25–35. Association for Computational Linguistics, Florence, Italy; 2019. https://doi.org/10.18653/v1/W19-3504. https://www.aclweb.org/anthology/W19-3504.

19. Davidson T, Warmsley D, Macy M, Weber I. 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, pp. 512–515; 2017.
20. Del Vigna F, Cimino A, Dell'Orletta F, Petrocchi M, Tesconi M. Hate me, hate me not: Hate speech detection on facebook. In: ITASEC; 2017.
21. Devlin J, Chang M, Lee K, Toutanova K. BERT: pre-training of deep bidirectional transformers for language understanding. CoRR arXiv:abs/1810.04805; 2018.
22. Devlin J, Chang MW, Lee K, Toutanova K. 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), pp. 4171–4186. Association for Computational Linguistics, Minneapolis, Minnesota; 2019. https://doi.org/10.18653/v1/N19-1423. https://www.aclweb.org/anthology/N19-1423.
23. Djuric N, Zhou J, Morris R, Grbovic M, Radosavljevic V, Bhamidipati N. Hate speech detection with comment embeddings. In: Proceedings of the 24th International Conference on World Wide Web, WWW '15 Companion, p. 29–30. Association for Computing Machinery, New York, NY, USA; 2015. https://doi.org/10.1145/2740908.2742760.
24. Do HTT, Huynh HD, Nguyen KV, Nguyen NLT, Nguyen AGT. Hate speech detection on vietnamese social media text using the bidirectional-lstm model; 2019. arXiv:1911.03648.
25. Dworkin R. A new map of censorship. Index Censorship. 2006;35(1):130–3. https://doi.org/10.1080/03064220500532412.
26. Gambäck B, Sikdar UK. Using convolutional neural networks to classify hate-speech. In: Proceedings of the First Workshop on Abusive Language Online, pp. 85–90. Association for Computational Linguistics, Vancouver, BC, Canada; 2017. https://doi.org/10.18653/v1/W17-3013. https://www.aclweb.org/anthology/W17-3013.
27. Gascó G, Rocha M, Sanchis-Trilles G, Andrés-Ferrer J, Casacuberta F. Does more data always yield better translations? In: Daelemans W, Lapata M, Màrquez L (eds.) EACL 2012, 13th Conference of the European Chapter of the Association for Computational Linguistics, Avignon, France, April 23-27, 2012, pp. 152–161. The Association for Computer Linguistics; 2012. https://www.aclweb.org/anthology/E12-1016/.
28. Gelashvili T. Hate Speech on Social Media: Implications of private regulation and governance gaps. Master's thesis, Lund University, Sweden; 2018.
29. Gelber K, McNamara L. Evidencing the harms of hate speech. Soc Identities. 2016;22:324–41.
30. Grave E, Bojanowski P, Gupta P, Joulin A, Mikolov T. Learning word vectors for 157 languages. In: Proceedings of the International Conference on Language Resources and Evaluation (LREC 2018); 2018.
31. Greevy E, Smeaton AF. Classifying racist texts using a support vector machine. In: Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '04, p. 468–469. Association for Computing Machinery, New York, NY, USA; 2004. https://doi.org/10.1145/1008992.1009074.
32. Gröndahl T, Pajola L, Juuti M, Conti M, Asokan N. All you need is "love": Evading hate speech detection. In: Proceedings of the 11th ACM Workshop on Artificial Intelligence and Security, AISec '18, p. 2–12. Association for Computing Machinery, New York, NY, USA; 2018. https://doi.org/10.1145/3270101.3270103.
33. Hern A. Revealed: catastrophic effects of working as a facebook moderator. The Guardian; 2019. https://www.theguardian.com/technology/2019/sep/17/revealed-catastrophic-effects-working-facebook-moderator. Accessed on 26 Apr 2020.
34. Heyman S. Hate speech, public discourse, and the first amendment. In: Hare I, Weinstein J (eds.) Extreme Speech and Democracy. Oxford Scholarship Online; 2009. https://doi.org/10.1093/acprof:oso/9780199548781.003.0010.
35. Hom C. A puyyle about pejoratives. Philos. Stud. 2012;159:383–405. https://doi.org/10.1007/s11098-011-9749-7.
36. Huynh TV, Nguyen VD, Nguyen KV, Nguyen NLT, Nguyen AGT. Hate speech detection on vietnamese social media text using the bi-gru-lstm-cnn model. arXiv:1911.03644; 2019.
37. Immpermium: Detecting insults in social commentary. https://kaggle.com/c/detecting-insults-in-social-commentary. Accessed on 27 Apr 2020.
38. Isasi AC, Juanatey A. Hate speech in social media: a state-of-the-art review; 2017.
39. Jiao X, Yin Y, Shang L, Jiang X, Chen X, Li L, Wang F, Liu Q. Tiny{bert}: Distilling {bert} for natural language understanding; 2020. https://openreview.net/forum?id=rJx0Q6EFPB.
40. Joulin A, Grave E, Bojanowski P, Mikolov T. Bag of tricks for efficient text classification. In: Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers, pp. 427–431. Association for Computational Linguistics, Valencia, Spain; 2017. https://www.aclweb.org/anthology/E17-2068.
41. Kim JY, Ortiz C, Nam S, Santiago S, Datta V. Intersectional bias in hate speech and abusive language datasets; 2020.
42. Kirch W. (ed.) Pearson's Correlation Coefficient, pp. 1090–1091. Springer Netherlands, Dordrecht; 2008. https://doi.org/10.1007/978-1-4020-5614-7_2569.
43. Kumar R. Ojha AK., M.S., M., Z.: Benchmarking aggression identification in social media. In: Proceedings of the First Workshop on Trolling, Aggression and Cyberbullying (TRAC-2018), pp. 1–11; 2018.
44. Kwok I, Wang Y. Locate the hate: Detecting tweets against blacks. In: Proceedings of the Twenty-Seventh AAAI Conference on Artificial Intelligence, AAAI'13, p. 1621–1622. AAAI Press; 2013.
45. Lan Z, Chen M, Goodman S, Gimpel K, Sharma P, Soricut R. Albert: A lite bert for self-supervised learning of language representations. In: International Conference on Learning Representations; 2020. https://openreview.net/forum?id=H1eA7AEtvS.
46. Liaw A, Wiener M, et al. Classification and regression by randomforest. R NEWS. 2002;2(3):18–22.
47. Liu Y, Ott M, Goyal N, Du J, Joshi M, Chen D, Levy O, Lewis M, Zettlemoyer L, Stoyanov V. RoBERTa: a robustly optimized BERT pretraining approach. arXiv:1907.11692; 2019.
48. MacAvaney S, Yao HR, Yang E, Russell K, Goharian N, Frieder O. Hate speech detection: Challenges and solutions. PLoS ONE. 2019;14(8):1–16. https://doi.org/10.1371/journal.pone.0221152.
49. Mandl T, Modha S, Mandlia C, Patel D, Patel A, Dave M. HASOC - hate speech and offensive content identification in indo-european languages. https://hasoc2019.github.io/call_for_participation.html. Accessed 20 Sept 2019.
50. Mandl T, Modha S, Patel D, Dave M, Mandlia C, Patel A. Overview of the HASOC track at FIRE 2019: Hate Speech and Offensive Content Identification in Indo-European Languages). In: Proceedings of the 11th annual meeting of the Forum for Information Retrieval Evaluation; 2019.
51. Matsuda MJ. Public response to racist spech: Considering the victim's story. In: R.D. M. J. Matsuda C. R. Lawrence III, K. Williams (eds.) Words that wound: Critical race theory,

assaultive speech, and the first amendment, pp. 17–52. Routledge, New York; 1993.

52. McHugh M. Interrater reliability: the kappa statistic. Biochemia medica : časopis Hrvatskoga društva medicinskih biokemičara / HDMB. 2012;22:276–82. https://doi.org/10.11613/BM.2012.031.

53. McLachlan GJ. Discriminant analysis and statistical pattern recognition, vol. 544. Amsterdam: Wiley; 2004.

54. Mehdad Y, Tetreault J. Do characters abuse more than words? In: Proceedings of the SIGDIAL2016 conference, pp. 299–303; 2016. https://doi.org/10.18653/v1/W16-3638.

55. Mikolov T, Grave E, Bojanowski P, Puhrsch C, Joulin A. Advances in pre-training distributed word representations. In: Proceedings of the International Conference on Language Resources and Evaluation (LREC 2018); 2018.

56. Mikolov T, Sutskever I, Chen K, Corrado G, Dean J. Distributed representations of words and phrases and their compositionality. In: Proc. NIPS, pp. 3111–3119; 2013.

57. Mondal M, Silva LA, Benevenuto F. A measurement study of hate speech in social media. In: Proceedings of the 28th ACM Conference on Hypertext and Social Media, HT '17, p. 85–94. Association for Computing Machinery, New York, NY, USA; 2017. https://doi.org/10.1145/3078714.3078723.

58. Müller K, Schwarz C. Fanning the flames of hate: Social media and hate crime. SSRN Electronic Journal; 2017. https://doi.org/10.2139/ssrn.3082972.

59. Nina-Alcocer V. Vito at HASOC 2019: Detecting hate speech and offensive content through ensembles. In: Mehta P, Rosso P, Majumder P, Mitra M (eds.) Working Notes of FIRE 2019 - Forum for Information Retrieval Evaluation, Kolkata, India, December 12-15, 2019, *CEUR Workshop Proceedings*, vol. 2517, pp. 214–220. CEUR-WS.org; 2019. http://ceur-ws.org/Vol-2517/T3-5.pdf.

60. Nobata C, Tetreault J, Thomas A, Mehdad Y, Chang Y. Abusive language detection in online user content. In: Proceedings of the 25th International Conference on World Wide Web, WWW '16, p. 145–153. International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, CHE; 2016. https://doi.org/10.1145/2872427.2883062.

61. Nourbakhsh, A., Vermeer, F., Wiltvank, G., van der Goot, R.: sthruggle at SemEval-2019 task 5: An ensemble approach to hate speech detection. In: Proceedings of the 13th International Workshop on Semantic Evaluation, pp. 484–488. Association for Computational Linguistics, Minneapolis, Minnesota, USA (2019). https://doi.org/10.18653/v1/S19-2086. https://www.aclweb.org/anthology/S19-2086.

62. Ohieku A, Sabo S. Journalism practice in an era of unguided utterances: framing of hate speech in selected Nigerian newspapers. Univ Nigeria Interdiscip J Commun Stud. 2020;24(1):129–40.

63. O'Regan C. Hate Speech Online: an (Intractable) Contemporary Challenge? Current Legal Problems. 2018;71(1):403–29. https://doi.org/10.1093/clp/cuy012.

64. Otter DW, Medina JR, Kalita JK. A survey of the usages of deep learning for natural language processing. IEEE Transactions on Neural Networks and Learning Systems pp. 1–21; 2020.

65. Park J, Fung P. One-step and two-step classification for abusive language detection on twitter. In: ALW1: 1st Workshop on Abusive Language Online; 2017.

66. Pedro Alonso Rajkumar Saini GK. TheNorth at HASOC 2019 Hate Speech Detection in Social Media Data. In: Proceedings of the 11th annual meeting of the Forum for Information Retrieval Evaluation; 2019.

67. Pennington J, Socher R, Manning CD. Glove: Global vectors for word representation. In: Proc. EMNLP, pp. 1532–1543; 2014. http://www.aclweb.org/anthology/D14-1162.

68. Pereira-Kohatsu JC, Sánchez LQ, Liberatore F, Camacho-Collados M. Detecting and monitoring hate speech in twitter. Sensors. 2019;19(21):4654. https://doi.org/10.3390/s19214654.

69. Popa-Wyatt M, Wyatt J. Slurs, roles and power. Philos Stud. 2018;175:2879–906. https://doi.org/10.1007/s11098-017-0986-2.

70. Raehtka A. Recognizing the evolution of racial slurs. Democrat & Chronicle; 2014. https://eu.democratandchronicle.com/story/editorial/2014/01/29/recognizing-the-evolution-of-racial-slurs/5017955/. Accessed on 2020-04-28.

71. Ross B, Rist M, Carbonell G, Cabrera B, Kurowsky N, Wojatzki M. Measuring the reliability of hate speech annotations: The case of the european refugee crisis. In: Beißwenger M, Wojatzki M, Zesch T (eds.) Proceedings of NLP4CMC III, pp. 6–9; 2016. https://doi.org/10.17185/duepublico/42132.

72. Saha K, Chandrasekharan E, De Choudhury M. Prevalence and psychological effects of hateful speech in online college communities. In: Proceedings of the 10th ACM Conference on Web Science, WebSci '19, p. 255–264. Association for Computing Machinery, New York, NY, USA; 2019. https://doi.org/10.1145/3292522.3326032.

73. Salminen J, Almerekhi H, Milenkovic M, JUNG SG, KWAK H, JANSEN BJ. Anatomy of online hate: developing a taxonomy and machine learning models for identifying and classifying hate in online news media; 2018.

74. Sanh V, Debut L, Chaumond J, Wolf T. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter; 2019.

75. Sap M, Card D, Gabriel S, Choi Y, Smith NA. The risk of racial bias in hate speech detection. In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pp. 1668–1678. Association for Computational Linguistics, Florence, Italy; 2019. https://www.aclweb.org/anthology/P19-1163.

76. Schapire RE, Singer Y. Improved boosting algorithms using confidence-rated predictions. Mach Learn. 1999;37(3):297–336.

77. Seganti A, Sobol H, Orlova I, Kim H, Staniszewski J, Krumholc T, Koziel K. Nlpr@srpol at semeval-2019 task 6 and task 5: Linguistically enhanced deep learning offensive sentence classifier. In: SemEval@NAACL-HLT; 2019.

78. Sun C, Qiu X, Xu Y, Huang X. How to fine-tune bert for text classification? arXiv:1905.05583; 2020.

79. Sundararajan M, Taly A, Yan Q. Axiomatic attribution for deep networks. CoRR arXiv:1703.01365; 2017.

80. Topidi K. Words that Hurt (2): National and International Perspectives on Hate Speech Regulation; 2019.

81. Ullmann S, Tomalin M. Quarantining online hate speech: technical and ethical perspectives. Ethics and Information Technology; 2019. https://doi.org/10.1007/s10676-019-09516-z.

82. Van den Rul C. Why have resolutions of the un general assembly if they are not legally binding? 2020.

83. Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, Kaiser Ł, Polosukhin I. Attention is all you need. Advances in Neural Information Processing Systems **2017-Decem**(Nips), 5999–6009; 2017.

84. Wang B, Ding Y, Liu S, Zhou X. Ynu_wb at HASOC 2019: Ordered neurons LSTM with attention for identifying hate speech and offensive language. In: Mehta, P, Rosso, P, Majumder P, Mitra M (eds.) Working Notes of FIRE 2019 - Forum for Information Retrieval Evaluation, Kolkata, India, December 12-15, 2019, CEUR Workshop Proceedings, vol. 2517, pp. 191–198. CEUR-WS.org; 2019. http://ceur-ws.org/Vol-2517/T3-2.pdf.

85. Warner W, Hirschberg J. Detecting hate speech on the world wide web. In: Proceedings of the Second Workshop on Language in Social Media, pp. 19–26. Association for Computational

Linguistics, Montréal, Canada; 2012. https://www.aclweb.org/anthology/W12-2103.

86. Waseem Z. 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, pp. 138–142. Association for Computational Linguistics, Austin, Texas; 2016. https://www.aclweb.org/anthology/W16-5618.
87. Waseem Z, Hovy D. Hateful symbols or hateful people? predictive features for hate speech detection on twitter. In: Proceedings of the NAACL Student Research Workshop, pp. 88–93. Association for Computational Linguistics, San Diego, California; 2016. https://doi.org/10.18653/v1/N16-2013. https://www.aclweb.org/anthology/N16-2013.
88. Wei X, Lin H, Yang L, Yu Y. A convolution-lstm-based deep neural network for cross-domain mooc forum post classification. Information. 2017;8:92. https://doi.org/10.3390/info8030092.
89. Wiegand M, Siegel M, Ruppenhofer J. Overview of the germeval 2018 shared task on the identification of offensive language. In: Proceedings of the GermEval 2018 Workshop, pp. 1–11; 2018.
90. Wolf T, Debut L, Sanh V, Chaumond J, Delangue C, Moi A, Cistac P, Rault T, Louf R, Funtowicz M, Brew J. Huggingface's transformers: State-of-the-art natural language processing. arXiv:1910.03771; 2019.
91. Wright RE. Logistic regression. In: Grimm LG, Yarnold PR (eds.) Reading and understanding multivariate statistics. American Psychological Association; 1995.
92. Young T, Hazarika D, Poria S, Cambria E. Recent trends in deep learning based natural language processing; 2017. arXiv:1708.02709. Cite arXiv:1708.02709 Comment: Added BERT, ELMo, Transformer.
93. Yuan S, Wu X, Xiang Y. A two phase deep learning model for identifying discrimination from tweets. In: Pitoura E, Maabout S, Koutrika G, Marian A, Tanca L, Manolescu I, Stefanidis K (eds.) Proc. EDBT 2016, Bordeaux, France, March 15-16, 2016, Bordeaux, France, March 15–16, 2016, pp. 696–697. OpenProceedings.org; 2016. https://doi.org/10.5441/002/edbt.2016.92.
94. Zampieri M, Malmasi S, Nakov P, Rosenthal S, Farra N, Kumar R. Predicting the type and target of offensive posts in social media. 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), pp. 1415–1420. Association for Computational Linguistics, Minneapolis, Minnesota; 2019. https://doi.org/10.18653/v1/N19-1144. https://www.aclweb.org/anthology/N19-1144.
95. Zampieri M, Malmasi S, Nakov P, Rosenthal S, Farra N, Kumar R. Semeval-2019 task 6: Identifying and categorizing offensive language in social media (offenseval). In: Proceedings of the 13th International Workshop on Semantic Evaluation, pp. 75–86; 2019.
96. Zampieri M, Nakov P, Rosenthal S, Atanasova P, Karadzhov G, Mubarak H, Derczynski L, Pitenis Z, Çöltekin c. SemEval-2020 Task 12: Multilingual Offensive Language Identification in Social Media (OffensEval 2020). In: Proceedings of SemEval; 2020.
97. Zhang Y, Zhou X, Witt RM, Sabatini BL, Adjeroh D, Wong ST. Dendritic spine detection using curvilinear structure detector and lda classifier. Neuroimage. 2007;36(2):346–60.
98. Zhang Z, Luo L. Hate speech detection: A solved problem? the challenging case of long tail on twitter. Semantic Web Accepted; 2018. https://doi.org/10.3233/SW-180338.
99. Zhang Z, Robinson D, Tepper J. Detecting hate speech on twitter using a convolution-gru based deep neural network. In: Gangemi A, Navigli R, Vidal ME, Hitzler P, Troncy R, Hollink L, Tordai A, Alam M, editors. The Semantic Web. Cham: Springer; 2018. p. 745–60.
100. Zhu X, Vondrick C, Ramanan D, Fowlkes CC. Do we need more training data or better models for object detection? In: Bowden R, Collomosse JP, Mikolajczyk K (eds.) British Machine Vision Conference, BMVC 2012, Surrey, UK, September 3-7, 2012, pp. 1–11. BMVA Press (2012). https://doi.org/10.5244/C.26.80.
101. Zia T, Akram M, Nawaz M, Shahzad B, Abdullatif A, Mustafa R, Lali M. Identification of hatred speeches on twitter. In: Proceedings of 52nd The IRES International Conference, pp. 27–32; 2016.
102. Zimbardo PG. The human choice: individuation, reason, and order versus deindividuation, impulse, and chaos. In: Nebraska Symposium on Motivation 17, pp. 237–307; 1969.
103. Zimmerman S, Kruschwitz U, Fox C. Improving hate speech detection with deep learning ensembles. In: Proc. LREC). European Language Resources Association (ELRA), Miyazaki, Japan; 2018. https://www.aclweb.org/anthology/L18-1404.