Salminen *et al. Hum. Cent. Comput. Inf. Sci.* (2020) 10:1
https://doi.org/10.1186/s13673-019-0205-6

Human-centric Computing and Information Sciences

RESEARCH Open Access

Developing an online hate classifier for multiple social media platforms

Joni Salminen[1,2*], Maximilian Hopf[3], Shammur A. Chowdhury[1], Soon-gyo Jung[1], Hind Almerekhi[4] and Bernard J. Jansen[1]

*Correspondence: joolsa@utu.fi
[1] Qatar Computing Research Institute, Hamad Bin Khalifa University, Doha, Qatar
Full list of author information is available at the end of the article

Abstract

The proliferation of social media enables people to express their opinions widely online. However, at the same time, this has resulted in the emergence of conflict and hate, making online environments uninviting for users. Although researchers have found that hate is a problem across multiple platforms, there is a lack of models for online hate detection using multi-platform data. To address this research gap, we collect a total of 197,566 comments from four platforms: YouTube, Reddit, Wikipedia, and Twitter, with 80% of the comments labeled as non-hateful and the remaining 20% labeled as hateful. We then experiment with several classification algorithms (Logistic Regression, Naïve Bayes, Support Vector Machines, XGBoost, and Neural Networks) and feature representations (Bag-of-Words, TF-IDF, Word2Vec, BERT, and their combination). While all the models significantly outperform the keyword-based baseline classifier, XGBoost using all features performs the best (F1 = 0.92). Feature importance analysis indicates that BERT features are the most impactful for the predictions. Findings support the generalizability of the best model, as the platform-specific results from Twitter and Wikipedia are comparable to their respective source papers. We make our code publicly available for application in real software systems as well as for further development by online hate researchers.

Keywords: Online hate, Toxicity, Social media, Machine learning

Introduction

Online hate, described as abusive language [1], aggression [2], cyberbullying [3, 4], hatefulness [5], insults [6], personal attacks [7], provocation [8], racism [9], sexism [10], threats [11], or toxicity [12], has been identified as a major threat on online social media platforms. Pew Research Center [13] reports that among 4248 adults in the United States, 41% have personally experienced harassing behavior online, whereas 66% witnessed harassment directed towards others. Around 22% of adults have experienced offensive name-calling, purposeful embarrassment (22%), physical threats (10%), and sexual harassment (6%), among other types of harassment. Social media platforms are the most prominent grounds for such toxic behavior. Even though they often provide ways of flagging offensive and hateful content, only 17% of all adults have flagged harassing conversation, whereas only 12% of adults have reported someone for such acts [13].

Manual techniques like flagging are neither effective nor easily scalable [14] and have a risk of discrimination under subjective judgments by human annotators. Since an automated system can be faster than human annotation, machine learning models to automatically detect online hate have been gaining popularity and bringing researchers from different fields together [15].

Even though hate has been observed as a problem in multiple online social media platforms, including Reddit, YouTube, Wikipedia, Twitter, and so on [5, 7, 16–18], apart from a few exploratory studies [15, 19], *there is a lack of development and testing of models using data from multiple social media platforms*. Instead, studies tend to focus on one platform. This mono-platform focus is problematic because there are no guarantees the models that researchers develop generalize well across platforms. It is a reasonable assumption that developing a universal hate classifier could benefit from the information retrieved from various training sets and contexts. The mono-platform focus is particularly vexing, because the lack of a general hate classifier requires researchers and practitioners to "reinvent the wheel", meaning that each time carrying out online hate research (OHR) in a specific social media platform, a new classifier needs to be developed. This results not only in repetitive intellectual effort but also to "barriers of entry" for researchers who lack the skills for model development but would be interested in interpretative OHR. Furthermore, the lack of universal classifiers means that the results across studies and social media platforms are not easily comparable. In sum, the fragmentation of models and feature representations needlessly complicates hate detection across different platforms and contexts.

To address these concerns, we undertake the development of a cross-platform online hate classifier. Our model performs well for detecting hateful comments across multiple social media platforms, utilizes advanced linguistic features, namely, Bidirectional Encoder Representations from Transformers (BERT) (see "BERT" section), and is made available for further use and development by researchers and practitioners. While we do not claim to develop *the* universal classifier that solves all problems in online hate detection, our results are indicative of the promise that this line of work carries for the larger community of OHR and can be further built upon.

Our research questions (RQs) are:

- *RQ1* How well do different algorithms and feature representations perform for hate detection in multiple social media platforms?
- *RQ2* What are the most impactful features when predicting hate in multiple social media platforms?
- *RQ3* How well does a machine learning model learn linguistic characteristics in the hateful and non-hateful language in a cross-platform environment?

Literature review

Theoretical underpinnings of online hate

Several concepts are commonly associated with the definition of online hate in the literature. As a phenomenon, online hate is cross-disciplinary; it has been studied using multiple theoretical lenses and conceptual frameworks, including social psychology,

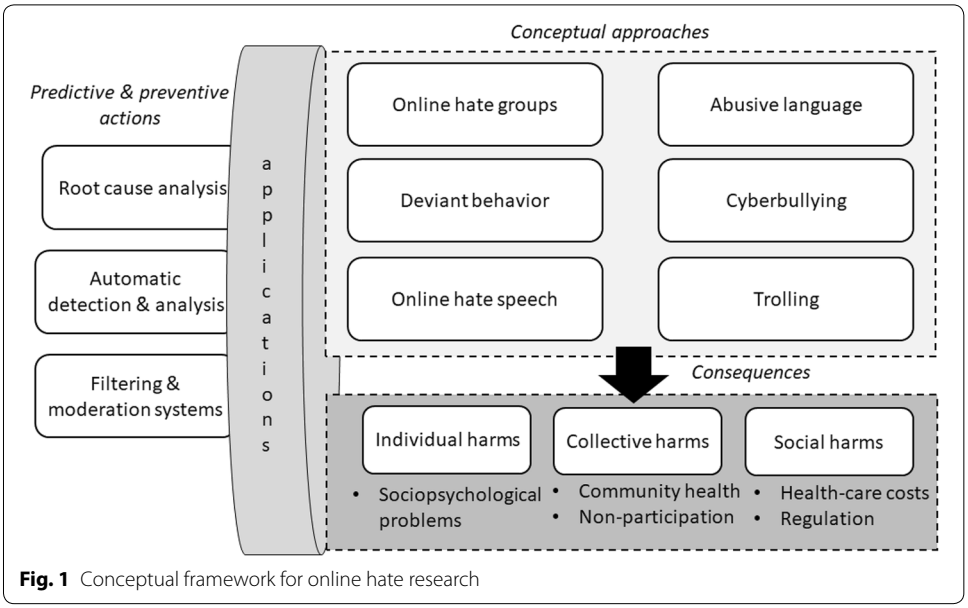

Fig. 1 Conceptual framework for online hate research

Human–Computer Interaction, politics, and legislation/regulative aspects. For example, Kansara et al. [20] present a framework for cyberbullying in social networks that contains harassment (i.e., sending offensive text messages and images), flaming (online violence using harsh messages), outing (personal information dissemination), exclusion (singling or leaving someone out of group), and masquerading (offensive communication using Sybil identities). Marret and Choo [21] present a framework of online victimization that highlights offline perpetration and parental conflict. These studies highlight the complex dynamics of online hate that complicate its automatic detection.

Figure 1 displays our conceptual framework of the focus areas in the extant OHR. First, online is seen as the use of *abusive, offensive, or profane language* [16, 22–25]. These studies tend to focus on the language aspects of online hate, such as linguistic styles, vocabularies, and ways of expression. Some of these studies deal with "counterspeech", i.e., ways of defusing the hateful comments with language-based strategies [26–28].

Second, some studies focus particularly on online hate as *hate speech* [9, 16, 29–31], i.e., "offensive post, motivated, in whole or in part, by the writer's bias against an aspect of a group of people. [underlining by us]" [32] (p. 87). The focal dimension here is targeting; i.e., the hate has a specific target such as refugees, women, a race, or religion [5, 33–35]. Waseem et al. [36] distinguish between different types of abuse segmented by the target of the abuse directed towards an individual/entity or generalized towards a group and the degree to which it is explicit. ElSherief et al. [37] study the relationship of hate instigators and targets and online visibility, finding that high-profile social media users attract more hate. Salminen et al. [5] find media and police to be major targets of hate in online news commenting. Overall, news-related discussions have been considered as a major hotbed for online toxicity [38].

Third, another important aspect of OHR is the consideration of group dynamics, visible in the studies focused on online hate groups and group prejudice [39], persuasive storytelling as hate conditioning [40], radicalization via social media extremist content

Table 1 Definitions of online hate

Definition of online hate	Source	Focus
"Language that is used to expresses hatred towards a targeted group or is intended to be derogatory, to humiliate, or to insult the members of the group"	Davidson et al. [16] (p. 215)	Language, target
"Hateful comments toward a specific group or target"	Salminen et al. [5] (p. 330), adapted from [109]	Target, group
"[Hate speech is] either 'directed' towards a specific person or entity, or 'generalized' towards a group of people sharing a common protected characteristic"	ElSherief et al. [67] (p. 1)	Target, group
"Comments that are rude, disrespectful or otherwise likely to make someone leave a discussion"	Almerekhi et al. [73], adapted from Jigsaw's toxic comment classification challenge in Kaggle[a]	Individual, comments, consequences
"An offensive post, motivated, in whole or in a part, by the writer's bias against an aspect of a group of people"	Mondal et al. [32] (p. 87)	Language, group, target
Offensive name calling, purposefully embarrassing others, stalking, harassing sexually, physically threatening, and harassing in a sustained manner	Wulzyn et al. [7], adapted from Pew Research Center	Language

[a] https://www.kaggle.com/c/jigsaw-toxic-comment-classification-challenge

[41], cultural transmission of hate [42], social exclusion [43], and so on. Due to a high degree of contextual and subjective factors, these nuances are often studied using interpretative methods.

Fourth, some studies focus on the *consequences of online hate* [12, 44, 45], meaning its effects on individuals and groups, for example, on the health of social media communities [46]. Often, these studies involve a predictive machine learning aspect for the detection and classification of toxicity in specific communities and social media platforms [2, 5, 16, 47, 48]. The central characteristic of toxicity studies is that they perceive online hate not only as the use of language but also as an *action having a concrete effect or outcome*. These outcomes may include the user leaving the toxic discussion [7], "silencing" or reduced participation in online social media [49], radicalization [40], group polarization where the previously held prejudices are enforced [50], degraded quality ("health") of an online community [2, 46, 51], offline violence [52] and security threats [53], and decreased feelings of safety and wellbeing of online users [33].

Finally, computer science studies in this field tend to focus on automating the detection of online hate. The positioning of this research falls within the computational stream of research, meaning experimentation with classifiers and features to improve automatic hate detection.

Definitions of online hate

Various authors in OHR cite the lack of commonly acknowledged definition for online hate [7, 16]. Instead of one shared definition, the literature contains many definitions with distinct approaches to online hate (see Table 1).

Building from this previous research (Table 1), our definition of online hate is as follows:

> *Online hate is composed of the use of language that contains either hate speech targeted toward individuals or groups, profanity, offensive language, or toxicity – in other words, comments that are rude, disrespectful, and can result in negative online and offline consequences for the individual, community, and society at large.*

This definition considers the central elements of the proposed definitions (Table 1): (a) use of language, (b) targeting, (c) individuals or groups, and (d) hate being action with consequences at various levels of society.

Evolution of online hate detection

Keyword-based classifiers

In general, the evolution of online hate detection can be divided into three temporal stages: (1) simple lexicon- or keyword-based classifiers, (2) classifiers using distributed semantics, and (3) deep learning classifiers with advanced linguistic features. An example of the first wave of studies is Sood et al. [6] that used a list of profane words, being able to identify 40% of words that are profane and then correctly identifying 52% as hateful/not hateful. Mondal et al. [32] used a simple sentence structure "I<intensity><userintent><hatetarget>" to identify hate targets. To avoid false positives, such as: "I really hate owing people favors", the researchers used specific hate targets (e.g., Black people, Mexican people, stupid people), and 1078 hate words from Hatebase,[1] a database of hateful words. In a similar vein, Davidson et al. [16] collected data from the Twitter API using the Hatebase lexicon as keywords. After the data was collected, crowd workers manually classified the 25K comments into hate speech, offensive speech, and neither. This categorization was done to ensure the keyword-based method resulted in hateful comments; however, the researchers found a significant mismatch between the hateful tweets and the crowd ratings [16], highlighting the deficiency of keyword searches for detecting hate. Salminen et al. [48] used a combination of manually tagged and keyword-detected training data to develop a neural network that removes hateful passages from comments that were tagged as hateful.

Despite their use in previous research, the limitations of using only keyword-based methods are well known. The main issue is the linguistic diversity of hate, which is not fully captured by a dictionary. For example, keywords cannot detect sarcasm and forms of humor [54]. Moreover, the dictionaries of hateful words and insults require constant updates [23], as new terminology and slang quickly develop in social media [6]. Furthermore, standards (i.e., *what* is interpreted as hate) differ by the online community, so that an expression that is hateful in one community may be considered neutral, humor, or typical discourse in another community [55]. For example, in online Q&A platforms like StackOverflow, a concise form of expression can appear rude to outsiders but be perfectly acceptable given the community standards [1].

Sahlgren et al. [56] point out another problem with keyword matching, namely, polysemy (i.e., the same word can have several different meanings). They mention the

[1] Structured repository of regionalized, multilingual hate speech: https://hatebase.org/.

Table 2 Challenges of online hate detection

Challenge	Description
False positive problem	False positives occur when a model detects a non-threatening expression as hateful content due to the presence of some words/phrases as a feature. For example, a tweet such as "Bill aims to fix sex-offender list's inequity toward gay men" can be labeled as hateful whereas, in reality, it is not an offensive expression but a simple statement
False negative problem	False negatives include cases when the model detects a threatening expression as non-threatening. For example, a keyword detector could correctly detect "I fucking hate Donald Trump", but ignore "Donald Trump is a rat". In reality, both of these expressions can be considered hateful
Subjectivity	The datasets can involve subjectivity arising from several sources. Crowd raters may not understand context or follow instructions. There can be high disagreement of what constitutes hate and various biases, such as racial bias [66, 110], can occur when constructing ground truth datasets. Sarcasm and humor further exacerbate the problem, as individuals' ability to interpret these types of language greatly varies
Polysemy	Polysemy, i.e., the same word or phrase having a different meaning in different contexts (e.g., social media community or platform) can greatly complicate the detection of online hate, as it introduces contextuality that the model should be aware of

example of "white trash" and "white trash cans", two instances with the same words but drastically different hate content. Another example is the word "fruit", which in general is non-abusive; however, when put in a specific context, the word can denote derogatory slang for a homosexual person. In Natural Language Processing (NLP), this problem is known as word-sense disambiguation [56], and it is considered highly challenging. Table 2 summarizes some of the challenges of developing online hate classifiers; these are extremely hard to solve with keyword-based methods alone.

Distributional semantics

While prior research indicates that keyword-based methods are not adequate to classify hate speech, language and words are necessary for the detection of online hate. For this reason, researchers have deployed a wide range of more sophisticated feature representations, including word n-grams, syntactic features, and distributional semantics (i.e., word embeddings and word vector space models). For example, Nobata et al. [23] detected hate speech, profanity, and derogatory language in social media using n-grams as well as linguistic, syntactic, and distributional semantics, finding that combining all feature sets yielded the best performance. Salminen et al. [5] used a similar set of features to detect hateful comments in a YouTube news channel, adding *term frequency* (TF), *term frequency—inverted document frequency* (TF-IDF), and word vectors. TF-IDF performed well in their study [5]. The study by Waseem et al. [36] suggested that features including mentions, proper nouns, other name entities, and co-reference resolution (finding all expressions that refer to the same entity in a text) can be useful for directed abuse detection. In turn, for generalized abuse, they suggested using independent vocabularies per target group to capture the lexical patterns. In cases of explicit abuse, both directed and generalized, the authors suggested the use of specific keywords along with polarity and sentiment as features for abuse detection. Djuric et al. [30] detected online hate using word embeddings from a neural network called *Paragraph2vec* to compare with the *Bag of Words* (BOW) model. In their study, Paragraph-2vec discovered some non-obvious swearing words and obtained better accuracy than BOW. Saleem et al. [55] used *Labeled Latent Dirichlet Allocation* (LLDA) to automatically

infer topics for the classifier, showing that Reddit communities have distinct linguistic practices that affect hate detection. Overall, distributional semantics relies on providing better representations of the hateful comments than keywords can do [56].

Deep learning classifiers

Some more recent work uses neural networks, particularly deep learning, for hate classification. These architectures, including variants of recurrent neural networks (RNN) [14, 24, 57], convolutional neural networks (CNN) [58], or their combination, produce state-of-the-art results. For example, Badjatiya et al. [29] classified the hatefulness of tweets using deep neural networks. They found that a CNN performed better than the baseline methods (character n-grams, TF-IDF, BOW). The best accuracy was obtained when combining deep neural networks with gradient boosted decision trees [29]. Park and Fung [59] detected racist and sexist language through a two-step approach with convolutional neural networks. They used three CNN models (CharCNN, WordCNN, and HybridCNN) on 20K tweets, achieving the best performance with HybridCNN and the worst with CharCNN. When two logistic regressions were combined, they performed as well as the one-step HybridCNN, and better than one-step logistic regressions [59]. Zhang et al. [58] used a pre-trained word embedding layer to map the text into vector space, which was then passed through a convolution layer with a max pooling downsampling technique. The output feature vector was then fed into a *Gated Recurrent Unit* (GRU) layer followed by global max pooling and a softmax layer. The previous studies indicate that CNNs' potential to capture the local patterns of features benefits online hate detection [1].

Finally, while most previous work relies on text features, there are also studies using other features. These include, for example, user features [60] and knowledge graphs [61]. For example, Chatzakou et al. [10] investigated user features (average posts, subscribed list, average session length), network features (the number of friends, followers, reciprocity), and authority, among others. In their study, the authors reported that they found network-based features to be more effective in aggressive user behavior classification. Similarly, Founta et al. [62] designed classifiers using both text and Twitter user metadata as features for the hate detection task. Their study shows that the best performance is obtained when combining the individual network trained jointly using the interleaved approach, indicating the usefulness of other features in addition to text. Qian et al. [63] found that intra-user (historical posts by the user) and inter-user (similar posts by other users) representations substantially improved the performance of hate speech detection.

Even though user features may improve the performance of hate detection, the scarcity of such information—i.e., user and network features are rarely available and not for all social media platforms—reduces researchers' ability to apply it for cross-platform hate detection.

Research gaps

Despite considerable previous OHR, cross-platform evaluation of online hate classifiers is typically omitted from research articles (see Table 3), even though it is well established that online hate is not restricted to a single platform or context.

Out of the existing studies that do use data from more than one platform, Silva et al. [35] evaluates their results on datasets from two platforms (Twitter and Whisper) and achieve reasonable performance ($F1 > 0.70$) on both datasets. The same combination

Table 3 Social media platforms often used in online hate detection research

Source	Primary source	Secondary source	Tertiary source
Hosseinmardi et al. [3]	Instagram	–	–
Almerekhi et al. [73]	Reddit	–	–
Kumar et al. [2]	Reddit	–	–
Davidson et al. [16]	Twitter	–	–
Silva et al. [35], Mondal et al. [32]	Twitter	Whisper	–
Badjatiya et al. [29]	Twitter	–	–
Chatzakou et al. [10, 111, 112]	Twitter	–	–
ElSherief et al. [37]	Twitter	–	–
Unsvag and Gambäck [60]	Twitter	–	–
Agarwal and Sureka [113]	YouTube	–	–
Salminen et al. [5]	YouTube	–	–
Djuric et al. [30]	Yahoo	–	–
Nobata et al. [23]	Yahoo	–	–
Wulczyn et al. [7]	Wikipedia	–	–

Typically, researchers use data only from one platform

of platforms was used by Mondal et al. [32]. Most typically, the use of the second platform is for evaluating the model developing using data from another platform. More rarely, data from several platforms is used for *both* model development and evaluation. Chandrasekharan et al. [64] is an exception, as they use comments from four platforms (4chan, MetaFilter, Reddit, and Voat) to predict hate on another platform (MixedBag). The best model of the researchers achieves accuracy of 90.20%, recall of 87.93% and precision of 91.09% [64]. Since this is one of the few models using data from more than two platforms and it is publicly available, we use it as a baseline model in our experiments (see "Experimental design" section). Regarding the use of multiple contexts, Karan and Snajder [19] studied the cross-domain use of abusive language in different online contexts, but they did not investigate hate between multiple online social media platforms. Park and Fung [59] used two datasets, but both were from Twitter. Mishra et al. [15] used Twitter and Wikipedia datasets for the detection of non-obvious hateful comments—as such, their approach did not consider general detection of *all* hateful comments.

Overall, the research on hate detection in multiple online social media platforms is scarce and, even the few studies that were published tend to find that the models are non-generalizable across domains [19]. Without cross-platform evaluation, the generalizability of models built on datasets from one online platform is restricted solely to that platform. Research efforts are needed for developing cross-platform online hate classifiers that are more universally applicable.

Furthermore, the replicability of the previous models is hindered by the fact that the resources—including code, algorithms, and datasets—are often not published, or authors promise to release but fail to do so [15]. We stress that there is an acute demand for online hate models that are made publicly available, so that organizations struggling with hateful online comments can leverage state-of-the-art research in their systems and operations. Fortunately, there are exceptions. For example,

Davidson et al. [16] make their code available on GitHub.[2] Our experiments show improvement over their results, as shown in "Experimental design and evaluation" section. Other research articles providing source code for hate detection model development and/or evaluation with links to code implementations that we could locate from our literature review include (implementations in footnotes) Waseem and Hovy [65],[3] Davidson et al. [66],[4] ElSherief et al. [67],[5] Saha et al. [68],[6] Qian et al. [69],[7] Ross et al. [70],[8] de Gibert et al. [71],[9] Badjatiya et al. [29],[10] and Chandrasekharan et al. [64].[11] However, none of these models specifically focus on cross-platform applicability.

Finally, continuous experimenting with NLP technologies and replicating their performance is important. We address these research gaps by (a) obtaining annotated training sets with ground-truth comments (i.e., known true labels based on human annotation) from four platforms: YouTube, Reddit, Wikipedia, and Twitter; (b) designing different classification models while exploring different feature representations and their combinations; and (c) evaluating the performance of the models for each platform separately and compare the outcomes with results reported in previous studies, as well as those obtained using a keyword-based classifier.

Datasets

Overview

We applied three criteria to select the datasets for this research: (a) the language is English, (b) the dataset was available at the time of conducting the research, and (c) the dataset and available details on the annotation procedure passed a manual evaluation (e.g., there was no high prevalence of false negatives/positives). Note that the previous research has found that online hate interpretation varies between individuals [72]. For this reason, OHR tends to apply aggregation methods such as majority vote, mean score, or consensus to determine if a comment is perceived as hateful or not. This precondition of "hateful on average" applies to all classifiers developed using this data.

In the following sections, we briefly explain each dataset and how they were merged into one online hate dataset. Note that different authors use different terminology when referring to hateful online comments (e.g., "toxic", "hateful", "abusive"). These terms may have some nuanced conceptual differences, but for this study, the definitions provided by the authors of the chosen datasets are aligned with our operational definition presented in "Introduction" section. In this research, we refer to all these comments as

[2] https://github.com/t-davidson/hate-speech-and-offensive-language.

[3] https://github.com/zeerakw/hatespeech.

[4] https://github.com/t-davidson/hate-speech-and-offensive-language.

[5] https://github.com/ben-aaron188/ucl_aca_20182019.

[6] https://github.com/punyajoy/Hateminers-EVALITA.

[7] https://github.com/jing-qian/A-Benchmark-Dataset-for-Learning-to-Intervene-in-Online-Hate-Speech.

[8] https://github.com/UCSM-DUE/IWG_hatespeech_public.

[9] https://github.com/aitor-garcia-p/hate-speech-dataset.

[10] https://github.com/pinkeshbadjatiya/twitter-hatespeech.

[11] https://bitbucket.org/ceshwar/bag-of-communities/src/master/.

hateful comments. When explaining the datasets, we will use the original authors' terms and then explain how their terms overlap with ours.

YouTube dataset (ICWSM-18-SALMINEN)

In this dataset, there are 3221 manually labeled comments posted on a YouTube channel of an online news and media company. Salminen et al. [5] note that many of the comments that are posted as reactions to the content in this channel are hateful, which makes the dataset promising for investigating online hate. The researchers used manual coding to annotate the data into hateful and non-hateful comments (as well as subsequent themes based on the target of the hate; however, this information is not used for our classifier). They provide detailed coding guidelines as well as inter-rater agreement measurement (agreement score = 75.3%), which the researchers interpret as substantial agreement. The agreement score was calculated by dividing the number of labels where two or more coders agreed by the number of possible values. The calculation was done for each coded item, and the item-based agreements were averaged to output the overall agreement. Overall, the dataset includes purposeful (i.e., intentionally hurtful) comments. This consideration was made because if hostility is not the purpose of the comment, it should not be classified as hateful. For example, "Trump is a bad president" was not considered as hateful, but "Trump is an orange buffoon" was considered as hateful. Also, the annotators considered linguistic patterns when annotating, such that swearing, aggressive comments, or mentioning past political or ethnic conflicts in a non-constructive and harmful way, were classified as hateful. When there was uncertainty about an instance, the researchers discussed it to avoid a biased label.

Reddit dataset (ALMEREKHI-19)

To detect toxicity triggers (i.e., causes) of online discussions in Reddit, the study of Almerekhi et al. [73] developed a model that detects the toxicity in the comments posted on Reddit communities (also denoted as subreddits). The dataset consists of relevance judgments specifying if a particular comment is hateful or not. Note that Almerekhi et al. [73] used the term "toxicity" as synonymous to "hateful". They selected for crowdsource labeling a random sample of 10,100 comments from *AskReddit* (one of the largest Reddit communities), which were obtained the *Pushshift API*. The designed labeling job asked workers to label a given comment as either toxic or non-toxic according to the toxicity definition provided by the *Perspective API*, which describes a toxic comment as "a rude, disrespectful, or unreasonable comment that is likely to make you leave a discussion".[12] The labeling results showed that 81.57% of the comments in the collection were labeled as non-toxic, while the remaining 18.43% were labeled toxic. The observed agreement between annotators was 0.85, and the Gwet's γ [51] was 0.70.

Wikipedia dataset (KAGGLE-18)

The *Wikipedia talk page Corpus* [7] includes datasets for three different categories: personal attack, toxicity, and aggression. The corpus is extracted from approximately 63 M

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

talk page comments processed from the public dump of the full history of English Wikipedia dated till 2015. For the annotation of each dataset, a random subset of comments was selected and annotated using *CrowdFlower*,[13] a crowdsourcing platform, with at least ten workers per comment. We refer to this dataset as "KAGGLE-18", because it was published by Jigsaw (a subsidiary of Alphabet, Google's parent company) as part of a Kaggle data science competition called "Toxic Comment Classification Challenge", available online at the time of writing.[14]

For this study, we used only the toxicity dataset consisting of 159,571 annotated comments, publicly available for download.[15] As is the case with the Reddit dataset, the Wikipedia dataset uses the term toxicity in the same sense as we use hatefulness in this work. For the toxicity task, the workers annotated comments based on the perceived toxicity and likelihood of making others leave the discussion. As each comment contains at least 10 judgments, a majority voting measure was used to assign the gold label, i.e., abusive or non-abusive. For our study, the abusive class was labeled as "hateful" and the non-abusive class was labeled as "non-hateful".

Twitter dataset (DAVIDSON-17-ICWSM)

This dataset is made available by Davidson et al. [16] who used crowd raters for labeling and provide a detailed description of the data collection principles. The dataset contains 25K tweets, randomly sampled from 85.4 M tweets extracted from the timeline of 33,458 Twitter users, using hate speech lexicon. The lexicon, compiled from Hatebase, contains words and phrases identified by internet users as hate speech. This lexicon was also used by the authors as keywords to extract the 85.4 M tweets. The selected 25K tweets were manually annotated by at least 3 workers using CrowdFlower. The task was to annotate the tweet with one of three categories: hate speech, offensive but not hate speech, or neither offensive nor hate speech. An agreement of 92% was obtained between the workers regarding the class labels for the task, and the final gold label for each tweet was assigned using a majority voting approach. The tweets with no majority class were discarded, making a total of 24,802 tweets with an assigned label of which 5% was given the hateful label. Note that the authors shared these tweets as "Tweet IDs", i.e., references to the original tweets. Therefore, we had to utilize the Twitter API to recollect the dataset. We were able to obtain 24,783 tweets (99.9% of the original dataset), with 19 tweets either deleted or otherwise unavailable. The less of only a small number of comments is unlikely to have a significant impact on the results when comparing our performance against that of Davidson et al.

Structuring the datasets into binary classes

Illustrating the datasets, Table 4 shows statistics on the datasets selected for classifier development. Table 5 shows the distribution of samples in the two classes in the aggregated dataset. Table 6 provides insights into the distribution of the classes in the test

[13] Currently known as Figure-Eight.

[14] https://www.kaggle.com/c/jigsaw-toxic-comment-classification-challenge.

[15] https://figshare.com/articles/Wikipedia_Detox_Data/4054689.

Table 4 Datasets combined for the study

Source	Platform and domain	Number of comments		Cum. count	% from total (%)
ICWSM-18-SALMINEN [5]	YouTube news media	T = 3221		3221	1.6
		H = 2364 (73.4%)	NH = 857 (26.6%)		
ALMEREKHI-19 [73]	Reddit 10 popular sub-communities	9991		13,212	5.1
		1619 (16.2%)	8372 (83.8%)		
DAVIDSON-17-ICWSM [16]	Twitter generic tweets	24,783		37,995	12.5
		20,620 (83.2%)	4163 (16.8%)		
KAGGLE-18 [7]	Wikipedia editor discussions	159,571		197,566	80.8
		15,294 (9.6%)	144,277 (90.4%)		

Breakdown under number of comments shows *T* total comments, *NH* non-hateful comments, *H* hateful comments

Table 5 Samples of the binary hate classes

Hateful (H)	Not hateful (NH)	Total
39,897	157,669	197,566
20.2%	79.8%	100%[a]

[a] Ratio of 75/25 (train/test) was used for model development

Table 6 Platform-wise description of test dataset used for the study

Platform	(% of full dataset)[a]	H-NH split (in %)
YouTube	798 (24.77%)	72.81–27.19
Reddit	2431 (24.33%)	16.25–83.75
Twitter	6290 (25.38%)	83.37–16.63
Wikipedia	39,873 (24.99%)	9.64–90.36
Total	49,392 (25.00%)	20.38–79.62

H represents Hateful comments and NH represents Not-Hateful comments. Test data was randomly sampled from each platform dataset

[a] Represents the number of instances in each group

data, grouped by each platform. Table 7 shows examples of hateful and non-hateful comments.

Note that we modified the datasets' original classes to match the binary classification. In Salminen et al. [5], the authors have two classes—neutral and hateful—to classify YouTube comments. We changed the first one to "not hateful". In Davidson et al. [16], the authors labeled tweets containing hate speech (we coded as hateful), offensive language (we coded as hateful), and neither (we coded as non-hateful). In a similar vein, Almerekhi et al. [73] use crowd raters with the majority voting to label toxic (hateful) and non-toxic (not hateful) Reddit comments. Similarly, Wulczyn et al. [7] use trained human raters to annotate toxic (hateful) and non-toxic (not hateful) Wikipedia comments.

Related datasets

In addition to the selected datasets, we considered the following datasets from prior literature: Waseem [9], Fortuna [74], Ross et al. [70], Salminen et al. [48], and the

Table 7 Examples of hateful comments from each platform

	Comment text	Platform
1	this is disgusting and heartbreaking. They should do the same shit to the cops of what they did to that poor guy. 10 s of pepper spray directly into the face, run over foot etc. equal it up a little bit, except for the detail of having a fucking stroke. So it still wouldn't be exactly what the guy went through. Fucking discusting. They get a hard on power tripping others. They are just fucking cowards	YouTube
2	Fuck that cop. Nothing but a fucking thug	YouTube
3	Fucking piece of shit cops, protect and serve my ass	YouTube
4	Land of the free… My fucking ass. America is one hell of a shit hole where its full of dumb sheeple believing in fictional fox news and cops who kill and terrorise more per day than terrorists in this whole world combined	YouTube
5	That stupid bitch Sally to just die. I know he's cheating on me with her, I've seen the texts on his phone. It's just a matter of time until she fucks up though!	Reddit
6	Yeah, you're not projecting at all… ??? This still doesn't make any sense… nobody even suggested your choice had anything whatsoever to do with lack of opportunity, lol. But whatever, it was obvious from the original comment I replied to you're not exactly a very rational human being, so, cheers bro	Reddit
7	Nope, I swallowed your moms pussy while she was snickering. The things I do for money but your dad was happy to pay it. To bad she was crying	Reddit
8	Old bitch:"Is this salmon gluten free?" Me: "ma'am gluten is a protein found is wheat and isn't present in any meat" Old bitch: "I ASKED YOU IF THIS SALMON HAS GLUTEN IN IT!" Me:"…….. no" Old bitch: "what good are you?"	Reddit
9	Nuke town that pussy	Twitter
10	Lovin hoes but thats same bitch we put the pipe in. She suckin dick, thats the same bitch you give goodnight kiss	Twitter
11	Old hoes get mad when you don't show them the attention they want	Twitter
12	RT @slimthugga: U a nurse 😂RT @Blaccstone: @slimthugga you a clown boy You music is trash…. I dare you say something back I bury your Bit&#8230	Twitter
13	All of my edits are good. Cunts like you who revert good edits because you're too stupid to understand how to write well, and then revert other edits just because you've decided to bear a playground grudge, are the problem. Maybe 1 day you'll realise the damage you did to a noble project	Wikipedia
14	You should do something nice for yourself, maybe go grab a couple of Horny Goat Weeds from your local convenience store and jack off for a little longer than 3 min tonight	Wikipedia
15	I'm sorry I screwed around with someones talk page. It was very bad to do. I know how having the templates on their talk page helps you assert your dominance over them. I know I should bow down to the almighty administrators. But then again, I'm going to go play outside…. with your mom	Wikipedia
16	Would you both shut up, you don't run wikipedia, especially a stupid kid	Wikipedia

StackOverflow dataset [1]. The Waseem dataset, containing 6910 tweets labeled for racism, sexism, or neither, was omitted because our manual inspection revealed a high rate of false positives. For example, the following comments were labeled as sexist by most of the raters in the Waseem dataset:

- "@FarOutAkhtar How can I promote gender equality without sounding preachy or being a 'feminazi'? #AskFarhan".
- "i got called a feminazi today, it's been a good day".
- "In light of the monster derailment that is #BlameOneNotAll here are some mood capturing pics for my feminist pals pic.twitter.com/3pTV0M9qOQ".

A similar analysis was done to the other datasets, but the same false positive problem was not found. Even though the nature of online hate has been found to

be subjective/interpretative [72, 75], in this case, the large number of false positive indicates that the raters are rating anything that *talks about sexism* as a sexist tweet, even when the tweets contain sarcasm or humor instead of hate. Possible reasons for a large number of false positives in this dataset are that the raters were given poor instructions, or the instructions were not properly understood. Waseem mentions the problem of false positives in his paper [9]. Here, many of the false positives seem to include sarcasm. Because the distinction between sarcasm/humor and true, intentional hate is crucial for OHR [6], and the purpose is defeated if the training data is not valid, we decided not to include this dataset in our study.

The dataset by Fortuna [74], made available through the *INESC TEC* repository, consists of 5668 Portuguese tweets. We omitted this dataset because we are focusing on the English language. The same rationale was applied for the dataset of Ross et al. [70] that contains tweets of refugee-related hate speech in the German language. While these two studies show that there is research on online hate in other languages, most of the current work and datasets are in English. Finally, the StackOverflow dataset, released in conjunction with the *2nd Workshop on Abusive Language Online*, is made accessible upon accepted application from the company. This dataset was omitted because previous research has shown it to be inconsistent [1]. This is because the dataset is based on a type of self-selection rather than an objective coding procedure and is thus subjected to social dynamics taking place in the social media platform in question—for example, some users may flag a reply as offensive to retaliate, while others do not bother to flag an offensive response at all. When annotated by independent coders, Castelle [1] found a high disagreement between the original flagging and the ratings of the independent coders. Finally, the dataset by Salminen et al. [48] was omitted because the authors used a keyword-based detection of hateful comments. As mentioned in the literature review, keyword-based techniques have been found be contain many sources of error [25, 55], again, risking misclassification of comments using sarcasm and humor, as well as lacking adaptability to the evolving use of language.

Classification algorithms

In this section, we discuss the classification algorithms. These were chosen based on the nature of the problem (i.e., binary classification) as well as their performance based on prior research.

Logistic regression (LR)

The choice of *logistic regression* (LR) is supported by its simplicity and common use for text classification [60]. Depending on the features, LR can obtain good results in online hate detection with low model complexity [76]. As such, including LR when comparing different models seems rational. Gunasekara and Nejadgholi [77] note that "conventional machine learning classifiers such as linear regressions models have also been used to effectively detect abusive online language." (p. 2). In addition to [31], LR has been used for online hate detection at least by Xiang et al. [78], Burnap and Williams [47], Waseem and Hovy [65], Davidson et al. [16], Wulczyn et al. [7], and Salminen et al. [5].

Naïve Bayes (NB)

Another traditional algorithm, often applied as a baseline in machine learning models [79], is the *Naïve Bayes* (NB) classifier. The algorithm is a simple probabilistic approach based on Bayes' theorem, with the conditional independence assumption, and the theorem of total probability. It calculates sets of probabilities by counting frequencies and combinations of values in the given dataset. Even though the assumption of conditional independence rarely holds in real-world data, the algorithm performs well in various supervised classification problems, including text analysis. In the context of online hate detection, NB has been applied at least by Dinakar et al. [80], Chen et al. [81], and Kwok and Wang [82]. Due to its commonness for text classification problems, it is logical to include NB in our experiments.

Support-vector machines (SVM)

Support-vector machines (SVM) is another algorithm commonly applied for text classification. The intuition of SVMs is to find a hyperplane that maximizes the marginal distance of the classes. The cases defining the hyperplane are called support vectors. In binary classification, the support vectors produce a hyperplane that divides the cases into two non-overlapping classes. At least the following works have experimented with SVM for online hate detection, with satisfactory results: Xu et al. [83], Dadvar et al. [84], Nobata et al. [23], and Salminen et al. [5]. The computational complexity of SVM is lower compared to deep learning models, and it provides more straight-forward interpretability [19]. For these reasons, including SVM into our experiments makes sense.

XGBoost

XGBoost (*Extreme Gradient Boosted Decision Trees*) is an ensemble algorithm that uses decision trees, i.e., structures that split the data into smaller subsets that divide the target. Trees are combined with gradient boosting to build successive models that learn from the previous models' errors. The models are penalized for growing too complex, thus helping generalization to new data. XGBoost also has a highly optimized implementation and includes a pairwise loss implementation, which makes it suitable for a wide range of classification problems [85]. Compared to the other classifiers, XGBoost is slightly rarer in the literature, although we could locate two previous studies using it in the online hate context [27, 68].

Feed-forward neural network (FFNN)

For designing the online hate classification model, we use a simple feed-forward neural network (FFNN) with two hidden layers of 128 and 64 hidden units followed by an output layer with sigmoid function. In each hidden layer, we use the *rectified linear unit* (ReLu) activation function [86]. We apply dropout of 0.20, batch size of 64, and we train the network for two epochs, determined with cross-validation (3 epochs were already overfitting, i.e., the validation error increased after 2 epochs[16]). The network uses the

[16] Another reason for this "small" number of epochs is that we are using a pre-trained network (BERT), which already has seen millions of batches before the finetuning is started. We only train 1 layer from scratch. As the dataset is quite big, two epochs are enough to fit these relatively simple functions.

Table 8 Simple features

Feature	Definition
Words	The number of words in the comment
Uppercase	The number of uppercase characters in the comment
Uppercase_per_word	The average number of uppercase characters in the words of the comment (i.e., number of uppercases divided by the number of words)
Punctuation	The number of punctuations such as a full stop, comma, or question mark used in the comment
Punctuation_per_word	The number of punctuation marks divided by the number of words in the comment
Numbers	The number of numbers in the comment
Numbers_per_word	The number of numbers divided by the number of words in the comment

Adam stochastic optimizer [87] along with binary cross-entropy to approximate the empirical distribution in the training set and the probability distribution defined by the model.

As explained earlier, there is a wide range of studies using deep-learning architectures for online hate classification [1, 24, 29]. An example is Park and Fung [59] who used a neural network for binary classification of online hate, i.e., similar to our approach. Overall, the FFNN serves to provide experimental results on the performance of a conventional neural network architecture on our feature sets, as we want to focus on simple architectures that can easily be deployed by researchers and software developers in the field alike. This excludes more complicated deep learning architectures from our comparison.

Feature representation

Simple features

Feature engineering and extraction are crucial steps in developing robust text classifiers [60]. For this reason, we experiment with various feature types with an increasing level of complexity.

To establish a baseline feature set, we compute some basic features from the comments. We surmise that simple features (see Table 8) like the length of the comment, use of uppercase characters and punctuation might help our classifiers because emotional statements are often written in caps lock, and lack punctuation or use it excessively. As our experiments reveal, these features do add information for the models (see "Experimental results" section).

Bag of words

BOW is a counting algorithm that encodes a sentence or a document in the dataset using a vector representation with |V| dimensions where |V| is the size of the selected vocabulary, V. The representation is unordered and encodes the weight of each member of the V, in the given instance, using various scoring techniques. In our case, for each word from a vocabulary, it counts the number of occurrences in the comment. We build the vocabulary on the training set and later use it to create the same BOW features on the test set. A list of English stop words (i.e., common words such as 'and', 'or' that do

not add information for the classification) is excluded because these words appear frequently in text and contain little information.

TF-IDF

Using TF-IDF, instead of simply counting the words, which would overemphasize frequent words, each word is weighted by its relative frequency. The TF-IDF features inform the model if a word appears more often in a comment than usually in the whole text corpus. Prior research has found TF-IDF features useful for online hate detection [5]. As with BOW, the TF-IDF vocabulary is built during the training of the model and then reused for the test set. Both BOW and TF-IDF are considered simple and proven methods for text classification [56].

Word embeddings

Word embeddings (also known as word vectors) are numerical representations of words that facilitate language understanding by mathematical operations [88]. Word embeddings rely on a vector space model that captures the relative similarity among individual word vectors, thereby providing information on the underlying meaning of words [89]. For this reason, word embeddings are widely used for text classification and online hate detection [5, 30, 32]. Previous research has shown that different pretrained word embeddings, including *fastText*, *Word2Vec,* and *GloVe* perform well for online hate detection [56], so the choice between the models can be seen as arbitrary. For this research, we chose the popular GloVe vectors from the *SpaCy*, a free, open-source library for NLP in Python.[17] Other research papers deploying GloVe for online hate detection include, for example, Mishra et al. [15] and Kshirsagar et al. [90] The GloVe model we apply is publicly available[18] and contains 685k keys and 20k unique vectors with 300 dimensions, trained on *Common Crawl* datasets.[19]

BERT

Transformers transform one sequence into another by eliminating any recurrence and replaces it with an attention mechanism to handle dependencies between the input and output of the system. With this architecture, a model can be trained more efficiently due to the elimination of sequential dependency on the previous words, increasing effectiveness for modeling long-term dependencies. A state-of-the-art model to utilize the encoder of the transformer is BERT [91], developed by a group of researchers from Google to improve state-of-the-art language representation. BERT has vastly outperformed previous models, such as the *Generative Pretrained Transformer* (GPT) [63] and *Embeddings from Language Models* (ELMo) [24], in tasks such as question answering [92], named-entity recognition [1], and natural language inference [76], among others. BERT has also achieved state-of-the-art results in online hate detection [92–96].

[17] https://spacy.io/.

[18] https://spacy.io/models/en#en_core_web_md.

[19] Common Crawl is a nonprofit organization that crawls the web and freely provides its archives and datasets to the public: commoncrawl.org.

The success of BERT is enabled by applying bidirectional training of transformers on large computational resources and data, unlike other models where the network was trained from left-to-right or combined left-to-right and right-to-left training sequences. The transformer encoder reads the entire sentence at once, therefore allowing the model to learn the context of a word based on its entire surroundings. Apart from bidirectionality, BERT also uses masking techniques in the input format before feeding the whole sentence to the transformer. Around 15% of the input words in a sentence are randomly replaced (masked) by a special token ([MASK]) or random words. The model is then optimized by predicting the masked words using the context from the non-masked words. To learn the relationship among two adjacent sentences, the training steps of the model also include a task of predicting if the second sentence, distinguished from the first sentence using end-of-sentence markers, is (not) subsequent of the first input sentence in the document. The model is optimized jointly by minimizing the loss of the two tasks mentioned above.

Even though training BERT is computationally expensive, once trained, its deployment for downstream text classification tasks (such as online hate detection) is straight-forward. Moreover, the finetuning process that we will apply before classifying is drastically less expensive than the full retraining of BERT (the trainable parameters are ~3 M, less than 3% of the total parameters of the network). Due to these favorable properties, we use a pre-trained BERT model ($BERT_{BASE}$) that has been trained by Google researchers using *Wikipedia* and *BookCorpus* datasets. The model we use has 12 layers, 768 hidden layers, and 110 M parameters and is freely available on GitHub.[20] We adopt this model to our hate classifiers using its TensorFlow implementation, finetuning the last 3 BERT layers and adding a fully connected 128-neuron layer and a 1-neuron output layer on top of the BERT model that predicts our comment label. We use this combination of the NN and the BERT model to update the final three layers of the BERT model during the optimization. After this, we can cut off the small neural network again and are left with a BERT model that is finetuned to our task. To be able to use BERT correctly, we preprocess our data the same way that the training data of the BERT model was treated, including the *WordPiece* tokenization for English.[21]

Experimental design and evaluation

Experimental design

In this study, we train different models using various algorithms (described in "Experimental design and evaluation" section) along with different feature representations (presented in "Discussion" section) and compare their performance. In addition to the different algorithm performance, we also evaluated the performance of the models using two baseline models: a keyword-based classifier (KBC). We choose KBC because it is a quick and easy approach for software developers to implement and, therefore, feasible in practice. We use the list of keywords developed by Salminen et al. [72]; this list contains 200 manually curated hateful phrases and is available online.[22] KBC checks if a

[20] https://tfhub.dev/google/bert_uncased_L-12_H-768_A-12/1.

[21] https://github.com/google-research/bert.

[22] https://github.com/joolsa/Binary-Classifier-for-Online-Hate-Detection-in-Multiple-Social-Media-Platforms.

comment contains hateful phrases defined in the dictionary and classifies the comment as hateful or non-hateful accordingly. We then compare the prediction of the system with the ground truth to calculate the performance. Second, we use the model by Chandrasekharan [64] that is publicly available as a downloadable Pickle file[23] and, equally importantly, was trained on data from several platforms, achieving solid performance when applied to an independent social media platform whose comments the model had not previously seen (accuracy = 75.27%, precision = 77.49%, and recall = 71.24%).

In addition to studying the importance of different feature representations along with different algorithms, we also evaluate and present the results of our two best trained classifiers for each targeted social media platform: Wikipedia, Twitter, Reddit, and YouTube. For comparison, we group our instances in the test set according to its platform (% of instances in each group along with its distribution of classes as shown in Table 6) and present the results accordingly. For completeness of the study, we also include the results previously published for each platform. However, due to differences in training/test distribution between the source papers and our work, the results are not entirely comparable.

Evaluation metrics

The classifier performance is measured using the test set (~25% of the total dataset) with two metrics: (a) F1 score and (b) receiver operating characteristic—area under the curve (ROC-AUC). The F1 score is the harmonic mean of precision and recall at a decision threshold of 0.50. The ROC computes precision and recall at all potential decision thresholds, so the area under the ROC curve is an appropriate metric to measure overall model performance. Equation 1 shows the formula for calculating the F1 score.

$$F_1 = 2 \times \frac{p \times r}{p + r} \tag{1}$$

where p = precision (i.e., positive predictive value) and r = recall (i.e., true positive rate). In this research, we only report the F1 measure along with ROC-AUC. In addition, the models are compared for statistically significant performance differences using McNemar's test with significance level $\alpha = 0.01$.

Experimental results

To evaluate different algorithms and features (RQ1), Table 9 (evaluation measure: F1) and Table 10 (evaluation measure: ROC-AUC) present the results obtained using different lexical feature representations and classification algorithms. Regarding the model families, XGBoost outperforms the other models on all feature sets except the BOW features, where the FFNN performs slightly better. XGBoost is closely followed by the FFNN on all other feature spaces. The other three model types (LR, NB, and SVM) perform worse on all feature subsets. On the different training sets, the performance of NB and LR ranks in different orders, while the SVM is always last. As expected, in baseline comparison, the KBC performed the worst.

[23] We use the "static model" available at https://bitbucket.org/ceshwar/bag-of-communities/src/master/.

Table 9 F1 scores (the highest scores italicized)

	Simple features	BOW	TF-IDF	Word2Vec	BERT	All features[a]
LR	0.062	0.764	0.768	0.828	0.891	0.892
NB	0.130	0.505	0.606	0.601	0.885	0.868
SVM	0.066	0.487	0.648	0.765	0.892	0.883
XGBoost	*0.400*	0.765	*0.774*	*0.880*	*0.916*	*0.924***
FFNN	0.064	*0.770*	0.769	0.847	0.893	0.894
KBC	n/a	n/a	n/a	n/a	n/a	0.388
BOC	n/a	n/a	n/a	n/a	n/a	0.084

** Significant at $p < 0.001$ (McNemar's test comparing predictions from XGBoost-BERT and XGBoost-All

[a] The features are concatenated into one big vector for each instance and used as input to the classifiers

Table 10 ROC-AUC scores (the highest scores italicized)

	Simple features	BOW	TF-IDF	Word2Vec	BERT	All features
LR	0.514	0.819	0.820	0.873	0.925	0.925
NB	0.524	0.738	0.809	0.761	0.938	0.934
SVM	0.515	0.661	0.74	0.818	0.924	0.911
XGBoost	*0.782*	0.932	*0.937*	*0.986*	*0.994*	*0.995*
FFNN	0.743	*0.934*	*0.937*	0.974	0.988	0.988

Comparing the feature representations, we observe a linear trend (see Fig. 2) in the performance of the classifiers when moving from simpler features to more advanced ones, with BERT giving the best results when comparing individual features. While the TF-IDF and BOW features perform much worse, their performance is still considerably higher than a random guess. The fact that TF-IDF models are only marginally better than the BOW models indicates that TF is not critical for the predictions. Likely, the most substantial information gain comes from the presence of certain words like "fuck", which can be detected by BOW features as well as by TF-IDF features.

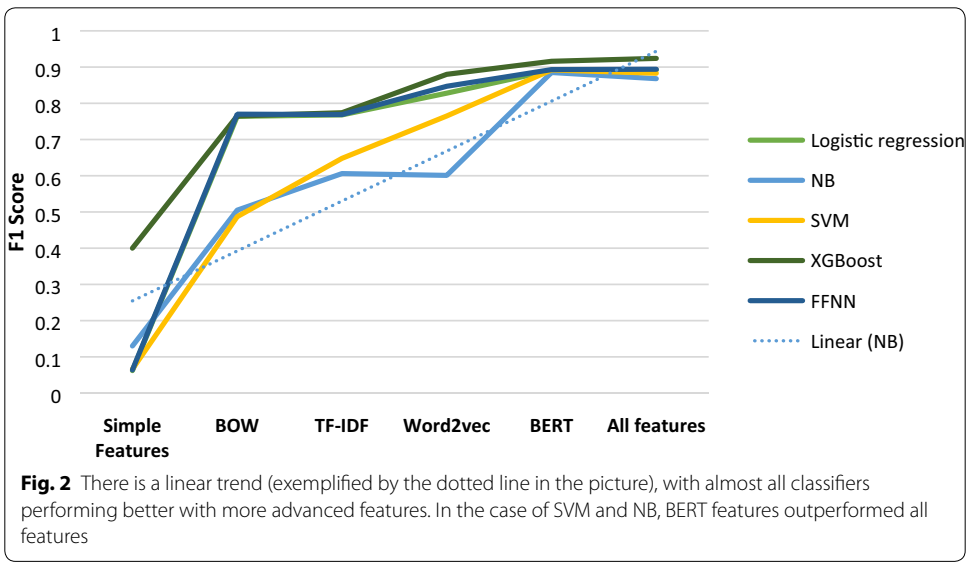

Fig. 2 There is a linear trend (exemplified by the dotted line in the picture), with almost all classifiers performing better with more advanced features. In the case of SVM and NB, BERT features outperformed all features

The results indicate that XGBoost outperforms the other algorithms most of the time, and XGBoost with all features is the highest performing model. This linearly combined feature set significantly outperforms XGBoost using only the BERT features. In contrast, the results using the FFNN shows no significant difference between the performance when using BERT only *versus* all features.

Baseline comparison shows that the KBC has drastically lower performance than any of the developed models, with an accuracy of 41.4% and F1 score of 0.388. The poor performance results from a high number of false positives (*Type I* error), which is 112,581 (57%) from the test set (recall = 0.25). In other words, the KBC considers many non-hateful comments as hateful, conforming with its known limitations [16]. Conversely, the problem of false negatives (*Type II* error) is much smaller for the KBC, as its precision is 0.919. This stems from the unbalanced dataset. In comparison, the false positive rate for the XGBoost model with all features is 2.0%, and the false negative rate is only 1.0%.

Surprisingly, the BOC model performs even worse than KBC, obtaining an F1 score of 0.084, precision of 0.085 and recall of 0.083. The accuracy is better than for the KBC (63.4% vs. 41.4%), but the results clearly indicate that the BOC model does not generalize well into these datasets. Unfortunately, we cannot test if *our* model generalizes to the original data of the BOC model [64], because the researchers are not sharing their data, only the Pickle file of the model. This also means that retraining using a sample of our data to improve their model is not possible. In terms of platform-specific performance, the BOC provides a better-than-chance (> 50%) accuracy for Wikipedia (70.4%) and Reddit (71.7%), but worse-than-chance accuracy for Twitter (19.6%) and YouTube (25.5%). The raw accuracy of the XGBoost model with all features is 97.0%, which implies a 94% improvement over a random model ($\kappa = (0.97 - 0.50)/(1 - 0.50) = 0.94$). Conversely, the performance over a random model is negative for the KBC (− 17.2%) and somewhat positive (+ 26.8%) for the BOC model.

Platform-specific analysis

For the platform-specific analysis (see Table 11), we use the XGBoost (All and BERT) models to predict the hatefulness of comments from each social media platform separately to assess the model's generalizability. The F1 score of XGBoost (All) outperforms XGBoost (BERT) significantly for Wikipedia and Twitter platforms; however, the same could not be said for the other two platforms. The features we use reflect the language being used in the social media platforms, so the difference in performance implies that the use of hateful language somewhat differs by platform (see "Linguistic variable analysis" section for more). Regarding the results, we consider the generalizability to be fair, as we achieve solid F1 and ROC-AUC scores (> 0.70) for each platform using XGBoost with BERT and all features (see Table 11).

Interestingly, the best model performs particularly well for YouTube ($F1_{xgboost\_all} = 0.91$) and Twitter ($F1_{xgboost\_all} = 0.980$). This implies the hateful language is easier to decipher for the model in these platforms. In contrast, the model performs worse with Reddit ($F1_{xgboost\_all} = 0.776$) and Wikipedia ($F1_{xgboost\_all} = 0.861$). On these two platforms, users may be more likely to engage in syntactically and semantically complex discussions, which makes it more difficult for the model (and perhaps for

Table 11 Generalizability of our best models (XGBoost with All and BERT features) across social media platforms

	YouTube	Reddit	Twitter	Wikipedia
$F1_{xgboost\_all}$	*0.911*	*0.776*	***0.980***	***0.861***
$F1_{xgboost\_BERT}$	*0.907*	***0.778***	*0.975*	*0.846*
F1 in original paper[a]	***0.960*** [5]	0.749 [73]	0.900 [16]	–
$ROC\text{-}AUC_{xgboost\_all}$	***0.968***	***0.967***	***0.994***	***0.993***
$ROC\text{-}AUC_{xgboost\_BERT}$	*0.964*	***0.967***	*0.991*	***0.991***
ROC-AUC in original paper[a]	–	0.957 [73]	–	0.972 [7]

We also present the results from previous research—when not reported, the cell contains (–). To facilitate reading, our results are given in italic. The results between the XGBoost (All vs BERT) in Wikipedia and Twitter platforms are significantly different with $p < 0.01$ (McNemar's test). Note that the results from previous studies are not directly comparable with the current study findings

[a] Brackets refer to sources

humans, too) to understand the hateful intent in their comments. Regarding the errors of the best model, many of the falsely classified comments can be seen difficult for a human to classify as well. For example, the comment "As usual, Jews and turks try to make famous Lebanese arab christian belong to them, he is Mexican Lebanese Arab christian and thats all!!" (from Wikipedia). This comment is labelled as 'not hateful' in the ground truth but the model classifies it as hateful because it probably detects a racist sentiment in the comment. Among the false negatives, there are many similar examples, such as "You WERE NOT REVERTING ALL THOSE TIMES EM. You deleted other contributions, every time something was added you didn't like. Get lost." (from Wikipedia). This comment is definitely annoyed or even angry, but it is not clear if it crosses the line of 'hateful'.

The highest risk for false positives using our model is in the Reddit platform (recall = 0.779). The lowest risk for false positives is when applied to the Twitter platform (recall = 0.978). The highest risk for missing hateful comments (false negatives) is again Reddit (precision = 0.813) and the lowest for Twitter (precision = 0.984). Overall, the model is slightly more likely to detect hateful comments when the comments are not hateful relative to classifying hateful comments as non-hateful (+ 3.9% relative difference).

We also analyze the possibility of overfitting by plotting the log loss of the XGBoost model on training and test sets. The model converges after about 75 trees (Fig. 3), after which the test error remains constant, indicating that there is little risk of overfitting.

Linguistic variable analysis

To investigate how well the best model (XGBoost with all features) learns linguistic characteristics in the hateful and non-hateful language (RQ3), we extracted scores on all linguistic variables available in the LIWC (Linguistic Inquiry and Word Count) [97] software. The LIWC taxonomy contains 93 categories that reflect the use of language at various levels, ranging from simple (word count, use of negations) to more complex (anxiety, tone) variables. To investigate the LIWC properties of the predicted comments, we applied the following procedure:

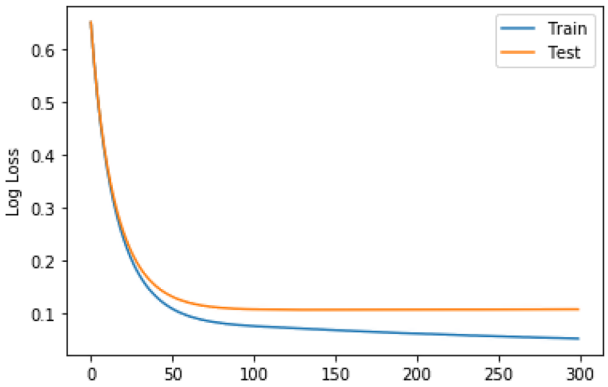

Fig. 3 Log loss of XGBoost (All features) model for training and test sets

1. Extract LIWC variable scores for all hateful and non-hateful comments in train and test set.
2. Create four comment sets:

   - $\text{hateful}_{\text{ground}}$ (containing comments whose ground truth value is hateful).
   - $\text{hateful}_{\text{predicted}}$ (predicted value == hateful).
   - $\text{non-hateful}_{\text{ground}}$ (ground truth value == non-hateful).
   - $\text{non-hateful}_{\text{predicted}}$ (predicted value == non-hateful).

3. Calculate the average score for each LIWC variable in each set.
4. Calculate relative difference D between average scores of ground truth and average scores of predicted comments (i.e., $(\text{hateful}_{\text{predicted}} - \text{hateful}_{\text{ground}})/\text{hateful}_{\text{ground}}$ and $(\text{non-hateful}_{\text{predicted}} - \text{non-hateful}_{\text{ground}})/\text{non-hateful}_{\text{ground}}$ for each LIWC variable.
5. Sort D by highest value and examine (a) which linguistic features are replicated well by the predictive model (i.e., their relative difference is small) and (b) which features are not well captured (i.e., their relative difference to ground truth is high) by the model.

Results (see Fig. 4) indicate that the model's predictions replicate the linguistic characteristics of both the hateful and non-hateful comments reasonably well (i.e., the difference scores are centered around zero). The average difference across all LIWC categories is M = 0.011 (SD = 0.240) for the Hateful paired comments and M = 0.002 (SD = 0.020) for the Non-hateful paired comments. Thus, hateful language is replicated more poorly relative to non-hateful language, and it has a considerably higher standard deviation among LIWC categories.

When examining the difference between predicted hateful comments and the ground truth hateful comments, out of the 93 LIWC categories, seven categories are classified as outliers (see Table 12). The predictions show more seldom use of (a) parentheses (− 13.7%, indicating less parentheses in predicted hateful than in ground truth hateful comments), (b) quotation signs (− 8.6%), (c) dashes (− 7.8%), and (d) question marks (− 5.6%). Moreover, the score for word count (WC) was 4.9%

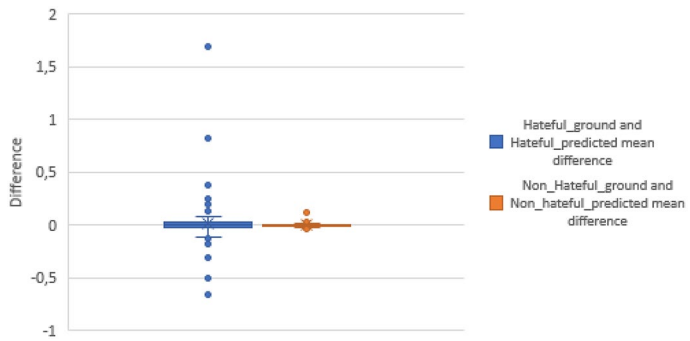

Fig. 4 Differences of aggregated mean scores of predicted labels' LIWC scores and ground-truth labels' LIWC scores

Table 12 Relative differences of linguistic variables between comments predicted as hateful by XGBoost + All and those labeled as hateful in the ground truth

LIWC category	Rel. diff. (lower scores) (%)	LIWC category	Rel diff. (higher scores) (%)
Parenth[a]	−13.4	Friend[a]	+6.9
Quote[a]	−8.6	Body[a]	+5.3
Dash[a]	−7.8	Swear[a]	+5.2
QMark	−5.6	Sexual[a]	+5.1
WC	−4.9	Bio	+4.7
Risk	−4.7	Informal	+4.6
anx	−4.5	Anger	+4.4
Work	−4.4	Semic	+4.0
Tone	−4.0	Netspeak	+3.6

Relative difference is calculated as $C_{predicted} - C_{ground})/C_{ground}$, where C is a LIWC category

[a] Outlier: > 1.5 interquartile ranges (IQRs) below the first quartile or above the third quartile

lower for predicted hateful comments relative to ground-truth hateful comments, which can be indicative of the model's capability to learn Twitter's short-messaging format well (see "Experimental results" section). In contrast, the predicted comments had a higher use of words from the Friends category (+6.9% relative to ground truth hateful comments)—this category contains, for example, references to 'pal', 'buddy', and 'coworker'. Similarly, the relatively higher scores for the Body (+5.3%; examples: 'ache', 'heart', 'cough'), Swear words (+5.2%), Sexuality (+5.1%; e.g., 'horny', 'love', 'incest'), and Anger (+4.4%; e.g., 'hate', 'kill', 'pissed') categories imply that the model is over-emphasizing the importance of their use when predicting hatefulness. Similarly, biological process (e.g., 'eat', 'blood', 'pain') and "netspeak", consisting of shorthand interpersonal communication (e.g., "lol", "4ever") [98], are also over-emphasized (+4.7% and +3.6%, respectively). For some reason, the use of semi-colons (SemiC) takes place more (+4.0%) in the predicted hateful comments than ground truth hateful comments.

Feature importance analysis

Addressing RQ2 (the impact of features on the predictions), we carry out a feature importance analysis by using Shapley values. Shapley values originate from game theory, where they are used to distribute a reward among the players in cooperative games [99]. When applying this concept to machine learning models, the game is the model accuracy, and the players are the different features. The important features, i.e., those that have a large influence on the model performance, will have large Shapley values.

Figure 5 shows the 30 most important features from test set predictions and their contribution to the predicted class. The dots with negative values on the x-axis are predictions where the specific feature had a negative contribution (less likely a hate comment) and vice versa. The most important takeaway is that out of the 30 most important features, 29 come from the BERT model. Only one other feature type (Word2Vec on the second to last row) is present. This outcome illustrates the importance of the BERT model for the classifier. Even though we only trained the last three layers of the model, it still produces better features than all the other approaches in this analysis. Unfortunately, the BERT features are not humanly interpretable, so it

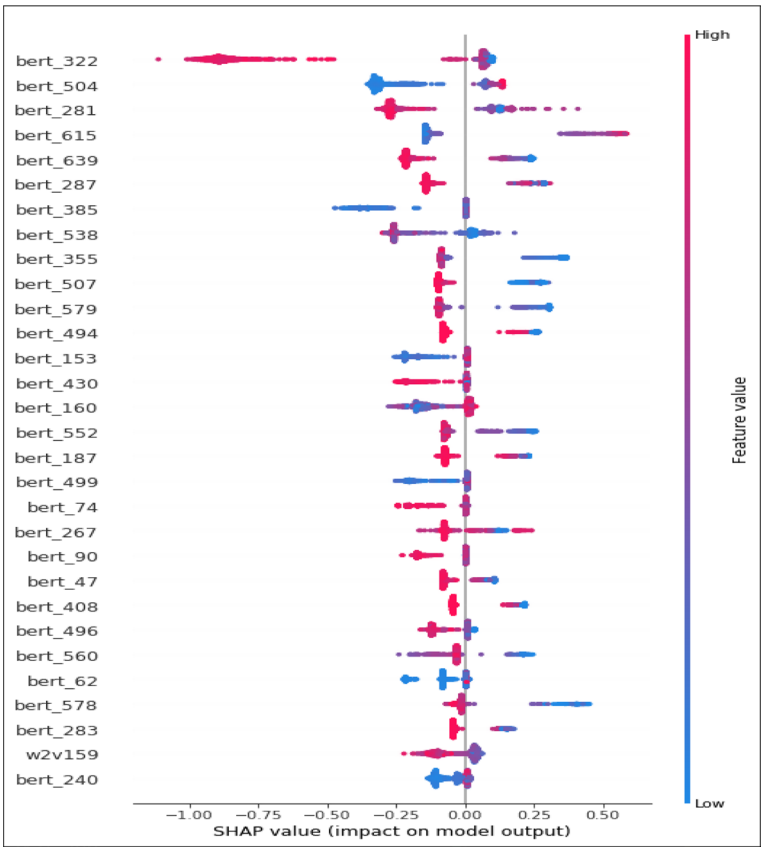

Fig. 5 Feature importance of the XGBoost model. The vertical axis represents the value of the feature, ranging from low to high. The horizontal axis represents the feature's impact on the model output. For example, a high value of "bert_322" (top-ranking feature, with the high value represented by red color) has a high negative impact across model predictions, with most SHAP values ranging between − 0.50 and − 1.00. The feature analysis shows the usefulness of BERT for online hate detection

Table 13 Most impactful words for hate prediction using LR and TF-IDF

Twitter		Reddit		YouTube		Wikipedia		All platforms	
Feature	**Coeff.**	**Feature**	**Coeff.**	**Feature**	**Coeff.**	**Feature**	**Coeff.**	**Feature**	**Coeff.**
tfidf_bitch	10.713	tfidf_fuck	8.937	tfidf_fuck	3.609	tfidf_fuck	14.174	tfidf_fuck	13.875
tfidf_bitches	8.800	tfidf_shit	7.932	tfidf_hate	3.341	tfidf_fucking	11.861	tfidf_bitch	12.983
tfidf_pussy	7.175	tfidf_fucking	7.633	tfidf_stupid	3.266	tfidf_shit	9.639	tfidf_fucking	11.188
tfidf_hoes	6.719	tfidf_dick	6.243	tfidf_fucking	3.257	tfidf_ass	8.467	tfidf_bitches	10.699
tfidf_hoe	5.185	tfidf_ass	4.733	tfidf_kill	2.684	tfidf_stupid	8.356	tfidf_shit	9.180

cannot be said as to why high values of, e.g. the *bert_322* feature are so strongly correlated with non-hateful comments.

For additional interpretability, we provide an analysis of LR results using TF-IDF, as this model performed the best (F1 = 0.768, see Table 9) out of the models that provide easily interpretable features (i.e., coefficients for individual words). Table 13 shows the most impactful terms for hate prediction using LR. The coefficients indicate the importance of a given feature for the models' predictions; a high coefficient implies that the feature is a strong predictor of hateful prediction.

Figure 6 shows the unions among the TOP15 hateful words of each platform (according to the LR classifier). On average, the unions contain 1.36 overlapping top hateful words. Top hateful words unique for Twitter mostly reflect racism and sexism (e.g., 'hoes', 'hoe', 'nigga'). Top hateful words unique for YouTube emphasize the news context and associated topics ('media', 'world', 'country'). Interestingly, for Reddit, the unique top hateful terms have the least signs of aggression when interpreted in isolation ('god', 'reading', 'people', 'seriously'). Hateful words in Wikipedia seem to coalesce with those in other platforms, as Wikipedia has only one unique word ('die') emerging from the analysis.

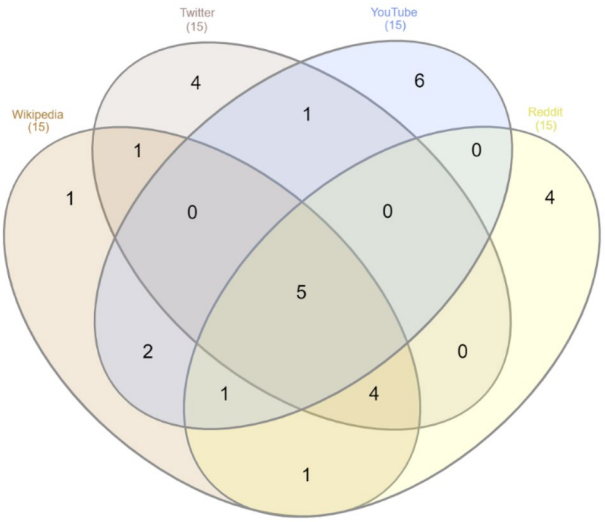

Fig. 6 Venn diagram of the TOP 15 hate predictor words from logistic regression

Discussion

Key contributions

Online hate is a rampant problem, with the negative consequence of prohibiting user participation in online discussions [100] and causing cognitive harm to individuals [101]. Since hate is prevalent across social media platforms, our goal was to develop a classifier that performs satisfactorily in multiple platforms. The results show that it is possible to train classifiers that can detect hateful comments in multiple social media platforms with solid performance and with the share of false positives and negatives remaining within reasonable boundaries. The fact that the models show their best performance when they are trained with BERT features supports the recent findings of bidirectional neural networks generating useful feature representations for online hate detection [92–96] [102]. Our results show a linear trend in the performance of the classifiers when moving from simpler features to more advanced ones, with BERT giving the best results.

In terms of novelty, our contribution is in the development and evaluation of online hate classifiers using data from *multiple* social media platforms, achieving satisfactory performance in each. As our analysis of overlapping hateful terms (see "Feature importance analysis" section) shows, hateful language bears similarity between the platforms, which explains why models can generalize in the first place. However, the communality of the hateful language also sets theoretical boundaries to generalizability, as deviations of what is considered hateful on average (or in a given context) would easily result in misclassification. Therefore, online communities where the use of language and the meaning of hatefulness deviates from the "mainstream" to a large extent are likely to require tailored efforts for hate detection also in the future. For these efforts, transfer learning using models such as BERT can be extremely helpful, as they improve the model's general understanding of language.

The implication of the hateful word analysis ("Feature importance analysis" section) is that our model is able to learn contextual nuances from different training sets and use this information for its predictions. This implies that, indeed, as proposed in the premise of this research, universal (or at least cross-platform applicable) hate classifiers can be developed, as models can learn specific hate vocabularies in different domains. As stated, the major risk in this is polysemy, as a specific word simultaneously loading high on hatefulness in Platform A and low in Platform B would confuse the learning. Even though investigating this tendency in cross-platform datasets is subject to future work, we note that such confusion could potentially be mitigated by using the source platform as a feature when training the model, as this may help contextualize the polysemic meanings of hateful words and hateful language in general.

Finally, to address the calls for openness [103], we are making the source code[24] of the classifiers public for further research and development and for software engineers to adopt in real systems and applications.

[24] https://github.com/joolsa/Binary-Classifier-for-Online-Hate-Detection-in-Multiple-Social-Media-Platforms.

Practical implications

Lack of open-sourced online hate classification models means that the results of the research papers are not easily replicable. For practitioners, it means that their access to high-quality classifiers is limited, and they are forced to use either (a) fallible KBC techniques that are easy to implement but—as shown by our experiments—perform poorly, or (b) black-box solutions provided by commercially backed organizations, such as Perspective API[25] developed by Jigsaw, a Google-backed organization. Since the technical details of these black-box solutions are unknown, they cannot be scrutinized or improved by the research community. This improvement is absolutely crucial since, as Silva et al. [35] demonstrate, the performance of hate detection models deteriorates over time. Without access to the critical resources (code, algorithms, and data), model retraining—and thus incremental progress in OHR—becomes cumbersome and unnecessarily challenging. Given the public nature of the hatefulness problem, these resources should be made public to the best ability of OHR scholars.

As an example, our classification model can be directly deployed to practical applications as well as further developed by other researchers. Regarding the use of the model in real systems (e.g., to automate moderation), we repeat the advice from a previous study [77] that essentially states that even small misclassification rates are a problem, as removing comments based on automatic detection methods can impact a user's freedom of speech in social media platforms [104]. It is highly unlikely that "perfect" classifiers for online hate would ever be developed, especially considering the subjective nature of what online hate is [72, 75]. Therefore, the machine learning models in this space should be considered as "helpers", or decision-making support tools, rather than the unequivocal truth. Their utility originates from the ability to quickly analyze a vast number of comments—way too many to manually inspect—and having the model detect the most likely hateful content for quarantining. Then, other users or moderators can make the final decision about the removal of such content, essentially applying human judgment and proper ethical considerations.

The prudent use of classifiers for automatic moderation is also important because the annotation of the hateful comments for model training is typically decontextualized, meaning it ignores community-specific ways of using language and different standards for what is hateful. As astutely remarked by Castelle [1], "*Because the flags are provided by users who have seen the entire interaction, many comments are considered offensive in context but not offensive when standing alone.*" (p. 8). For these reasons, ethical considerations in online hate detection are important [104]. Therefore, we do not advocate letting the model automatically decide on banning or removal of messages (perhaps apart from situations where false positives play only a smaller role). Rather, hate detection models can be used to flag comments for human moderation, following the *human-in-the-loop* paradigm [105]. This recommendation is also compatible with the "Zuckerberg principle" (see Fig. 7) to moderate hateful content in Facebook.

Finally, in practice, we recommend running the model using a GPU instead of CPU, as this increases the speed of predictions considerably, as shown in Fig. 8. Moreover,

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

"Some problems lend themselves more easily to A.I. solutions than others. So, hate speech is one of the hardest, because determining if something is hate speech is very linguistically nuanced. Right?... I'm optimistic that over a five to 10-year period, we will have A.I. tools that can get into some of the nuances, the linguistic nuances of different types of content to be more accurate in flagging things for our systems."

Fig. 7 Manual supervision, as exemplified by Facebook, is crucial for implementing hate detection algorithms in practice. (Source for picture (Creative Commons): http://pngimg.com/download/33386. Source for quote: [108] (p. 81))

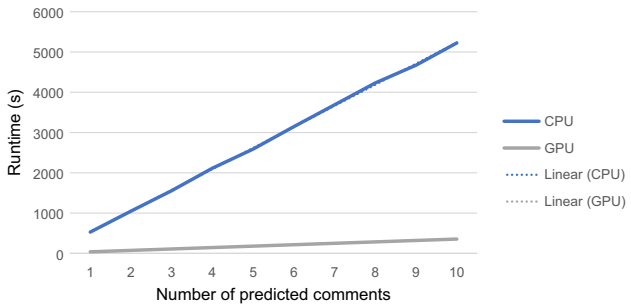

Fig. 8 Runtimes (averaged over 11 repetitions) of the XGBoost + All features model. Time complexity is linear, with GPU performing drastically better. The CPU was tested using a local machine (Intel Core i7-3770 @3.40 GHz), while the GPU was tested using a Tesla K80 running on Google Cloud

despite the BERT features being the most impactful for the predictions, other features are not useless either, as Word2Vec features had only a minor decrease in ROC-AUC. Therefore, software developers can strike a balance between accuracy and computational complexity by using word embeddings, while still achieving (possibly) acceptable results.

Limitations and future research

We identify several points of improvement for future research. First, our comparison of the classifier performance was made without hyperparameter optimization to ensure a fair baseline for each model family. However, the sensitivity of different algorithms for optimization varies, with SVMs and FFNN losing relatively more in comparison to other classifiers when hyperparameter optimization is not performed. Even though changing the experiment design to include hyperparameter optimization for all models could slightly change the order of the classifier performance, we do not expect that these changes would be drastic. Rather, XGBoost and FFNN would most likely continue having the highest performance. Nonetheless, this limitation leaves space for further improvement using different architectures and sets of parameters.

Second, there is a need for a detailed failure analysis of false positives and negatives to understand better where the model makes mistakes. Due to the scope of this manuscript, we are omitting a comprehensive analysis. Our indicative look at the results

suggests that even the best model struggles in "boundary cases" where also a human would struggle to determine if a comment is hateful or not (e.g., when more contextual information is needed to judge). Especially mitigating the problem of false positives is important because these can impact a user's reputation or freedom of speech. Additionally, a more detailed failure analysis could help to develop a more sophisticated classifier with several pipeline steps such as sarcasm detection, co-reference analysis, and so on. Again, this is left for future work.

Third, the binary classification in itself can be considered as a limitation. Previous research has shown that hate has a range of interpretations [72, 75], and understanding the context of the comments can be crucial for hate detection [106]. Instead of binary classification, some researchers have opted for identifying hate targets [5, 16] and more nuanced classes of online hate. While we consider the binary classification task as suitable within the scope of this study, we also acknowledge the need for the more nuanced online hate classifiers in the field of OHR. However, for the purpose of flagging hateful social media comments, our model provides high utility.

Fourth, even though there is a need to develop a single system that works across multiple platforms, the performance variation between the platforms implies that the patterns of hate have platform-specific distinctiveness. It is likely that there will always be nuanced linguistic differences in how people express hate across social media platforms, and also variation in how individuals experience hate [72]. Therefore, the efforts to develop one universal classifier will be limited by design. Future development efforts include user modeling to account for individual and group differences in hate interpretation as well as considering platform and community specificities when retraining models for increasing generalizability. In addition, continuous development and experimentation with novel NLP techniques are needed. For example, initial results with XLNet, a generalized autoregressive pretraining method [107], seem promising. Due to the scope of this research, we leave further experiments with these technologies for future work.

Finally, for future research to be successful towards the goal of 'as universal hate classifier as possible', enhanced efforts for sharing resources among researchers is needed. The reasons for not sharing can include exclusive research partnerships with firms and other organizations (e.g., [64]) or the source platform's terms of service (e.g., [5, 16]) that prohibit direct sharing of training data. While these reasons are understandable, the OHR community should seek ways to enhance the replicability of its results. Wider dissemination of resources, including the sharing of code, models, algorithms, and data is needed.

Conclusion

Online hate detection is needed to reduce toxicity in social media platforms. In this research, we experimented with various machine learning models (Logistic Regression, Naïve Bayes, Support-Vector Machines, XGBoost, and Neural Network) for online hate detection and found the best performance with XGBoost as a classifier and BERT features as the most impactful representation of hateful social media comments. The generalizability of the model to multiple social media platforms is good but varies slightly between the platforms. Our findings support the goal of developing more universal online hate classifiers for multiple social media platforms. The model that we make

publicly available can be deployed to practical applications as well as be further developed by online hate researchers.

Acknowledgements
The authors would like to thank Mridul Nagpal (International Institute of Information Technology) and Raghvendra Mall (Qatar Computing Research Institute) for discussions relating to online hate research, as well as Julia Silge (Stack Overflow) and Jennifer Golbeck (University of Maryland) for providing online hate datasets for review.

Authors' contributions
JS—creating the research plan, collecting and combining the datasets, designing the experiments, writing the research paper. MH—programming the experiments. SAC—evaluating the results and methodology, running additional analyses, editing the research paper. SJ—providing support for data collection and research plan. HA—providing support for data collection and research plan. BJJ—providing support for data collection and research plan. All authors read and approved the final manuscript.

Funding
The funding of the publication of this research was done by Qatar National Library (QNL).

Availability of data and materials
The data is made available upon request from the authors.

Competing interests
The authors declare that they have no competing interests.

Author details
[1] Qatar Computing Research Institute, Hamad Bin Khalifa University, Doha, Qatar. [2] University of Turku, Turku, Finland. [3] Pylink Ltd, London, UK. [4] Hamad Bin Khalifa University, Doha, Qatar.

Received: 3 July 2019 Accepted: 9 December 2019
Published online: 02 January 2020

References

1. Castelle M. The linguistic ideologies of deep abusive language classification. In: Proceedings of the 2nd workshop on abusive language online (ALW2), Brussels; 2018. P. 160–70
2. Kumar S, et al. Community interaction and conflict on the web. In: Proceedings of the 2018 world wide web conference on world wide web; 2018. P. 933–43
3. Hosseinmardi H et al (2015) Analyzing labeled cyberbullying incidents on the instagram social network. Soc Inf 2015:49–66
4. Wachs S et al (2019) Understanding the overlap between cyberbullying and cyberhate perpetration: moderating effects of toxic online disinhibition. Crim Behav Mental Health 29(3):179–188. https://doi.org/10.1002/cbm.2116
5. Salminen J, et al. Anatomy of online hate: developing a taxonomy and machine learning models for identifying and classifying hate in online news media. In: Proceedings of the international AAAI conference on web and social media (ICWSM 2018), San Francisco; 2018
6. Sood SO et al (2012) Automatic identification of personal insults on social news sites. J Am Soc Inform Sci Technol 63(2):270–285
7. Wulczyn E, et al. Ex Machina: personal attacks seen at scale. In: Proceedings of the 26th international conference on world wide web, Geneva; 2017. P. 1391–9
8. Mkono M (2018) 'Troll alert!': provocation and harassment in tourism and hospitality social media. Curr Issues Tour 21(7):791–804. https://doi.org/10.1080/13683500.2015.1106447
9. 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; 2016. P. 138–42
10. Chatzakou D, et al. Measuring #GamerGate: A tale of hate, sexism, and bullying. In: Proceedings of the 26th international conference on world wide web companion, Geneva; 2017. P. 1285–90
11. Willard NE (2007) Cyberbullying and cyberthreats: Responding to the challenge of online social aggression, threats, and distress. Research press, Champaign
12. Märtens M, et al. Toxicity detection in multiplayer online games. In: Proceedings of the 2015 international workshop on network and systems support for games, Piscataway; 2015. P. 5:1–5:6
13. Pew Research Center 2017. Online Harassment 2017
14. Pavlopoulos J, et al. Deeper attention to abusive user content moderation. In: Proceedings of the 2017 conference on empirical methods in natural language processing; 2017. P. 1125–35
15. Mishra P, et al. Neural character-based composition models for abuse detection. arXiv preprint arXiv:1809.00378. 2018
16. Davidson T, et al. Automated hate speech detection and the problem of offensive language. In: Proceedings of eleventh international AAAI conference on web and social media, Montreal; 2017. P. 512–5
17. Mohan S et al (2017) The impact of toxic language on the health of reddit communities. Springer, Berlin, pp 51–56
18. Watanabe H et al (2018) Hate speech on twitter: a pragmatic approach to collect hateful and offensive expressions and perform hate speech detection. IEEE Access. 6(2018):13825–13835. https://doi.org/10.1109/ACCESS.2018.2806394

19. Karan M, Šnajder J. Cross-domain detection of abusive language online. In: Proceedings of the 2nd workshop on abusive language online (ALW2); 2018. P. 132–137
20. Kansara KB, Shekokar NM (2015) A framework for cyberbullying detection in social network. Int J Curr Eng Technol 5:1
21. Marret MJ, Choo WY (2017) Factors associated with online victimisation among Malaysian adolescents who use social networking sites: a cross-sectional study. BMJ Open 7(6):e014959. https://doi.org/10.1136/bmjopen-2016-014959
22. Lee H-S et al (2018) An abusive text detection system based on enhanced abusive and non-abusive word lists. Decis Support Syst. https://doi.org/10.1016/j.dss.2018.06.009
23. Nobata C, et al. Abusive language detection in online user content. In: Proceedings of the 25th international conference on world wide web, Geneva, Switzerland; 2016. P. 145–53
24. Pitsilis GK, et al. Detecting offensive language in tweets using deep learning. arXiv preprint arXiv:1801.04433. 2018
25. Sood S, et al. Profanity use in online communities. In: Proceedings of the SIGCHI conference on human factors in computing systems, New York; 2012. P. 1481–90
26. Khorasani MM (2008) Controversies in online discussion forums. Fest-Platte für Gerd Fritz. 14:1
27. Mathew B, et al. Thou shalt not hate: countering online hate speech. In: Proceedings of the 13th international AAAI conference on web and social media (ICWSM-2019). Munich; 2019
28. Wright L, et al. Vectors for counterspeech on twitter. In: Proceedings of the first workshop on abusive language online; 2017. P. 57–62
29. Badjatiya P, et al. Deep learning for hate speech detection in tweets. In: Proceedings of the 26th international conference on world wide web companion, Geneva; 2017. P. 759–60
30. Djuric N, et al. Hate speech detection with comment embeddings. In: Proceedings of the 24th international conference on world wide web, New York; 2015. P. 29–30
31. Fortuna P, Nunes S (2018) A survey on automatic detection of hate speech in text. ACM Comput Surv 51:4. https://doi.org/10.1145/3232676
32. Mondal M, et al. A measurement study of hate speech in social media. In: Proceedings of the 28th ACM conference on hypertext and social media, New York; 2017. P. 85–94
33. Herring S et al (2002) Searching for safety online: managing"trolling"in a feminist forum. Inf Soc 18(5):371–384
34. Räsänen P et al (2016) Targets of online hate: examining determinants of victimization among young Finnish Facebook users. Viol Vict 31(4):708
35. Silva L, et al. Analyzing the targets of hate in online social media. In: Proceedings of tenth international AAAI conference on web and social media, Palo Alto; 2016
36. Waseem Z, et al. Understanding abuse: a typology of abusive language detection subtasks. arXiv:1705.09899[cs]. 2017
37. ElSherief M, et al. Peer to peer hate: hate speech instigators and their targets. In: Proceedings of the twelfth international AAAI conference on web and social media, Palo Alto; 2018
38. Qayyum A, et al. Exploring media bias and toxicity in south asian political discourse. In: 2018 12th international conference on open source systems and technologies (ICOSST); 2018. P. 1–8
39. Brewer MB (1999) The psychology of prejudice: ingroup love and outgroup hate? J Soc Issues 55(3):429–444
40. Lee E, Leets L (2002) Persuasive storytelling by hate groups online: examining its effects on adolescents. Am Behav Sci 45(6):927–957
41. Gerstenfeld PB et al (2003) Hate online: a content analysis of extremist Internet sites. Anal Soc Issues Public Policy 3(1):29–44
42. Hale L (2014) Globalization: cultural transmission of racism. Race Gender Class 21(2):112–125
43. Birk MV, et al. The effects of social exclusion on play experience and hostile cognitions in digital games. In: Proceedings of the 2016 CHI conference on human factors in computing systems, New York; 2016. P. 3007–19
44. Adinolf S, Turkay S. Toxic behaviors in Esports games: player perceptions and coping strategies. In: Proceedings of the 2018 annual symposium on computer–human interaction in play companion extended abstracts, New York; 2018. P. 365–72
45. Rodriguez N, Rojas-Galeano S. Shielding Google's language toxicity model against adversarial attacks. arXiv :1801.01828[cs]. 2018
46. Kwon KH, Gruzd A (2017) Is offensive commenting contagious online? Examining public vs interpersonal swearing in response to Donald Trump's YouTube campaign videos. Int Res 27(4):991–1010. https://doi.org/10.1108/IntR-02-2017-0072
47. Burnap P, Williams ML (2016) Us and them: identifying cyber hate on Twitter across multiple protected characteristics. EPJ Data Sci 5(1):11. https://doi.org/10.1140/epjds/s13688-016-0072-6
48. Salminen J, et al. Neural network hate deletion: developing a machine learning model to eliminate hate from online comments. In: Lecture notes in computer science (LNCS 11193), St. Petersburg; 2018
49. Bowler L et al (2015) From cyberbullying to well-being: a narrative-based participatory approach to values-oriented design for social media. J Assoc Inf Sci Technol 66(6):1274–1293. https://doi.org/10.1002/asi.23270
50. Del Vicario M et al (2016) Echo chambers: emotional contagion and group polarization on facebook. Sci Rep 6:37825. https://doi.org/10.1038/srep37825
51. Mossie Z, Wang J-H (2019) Vulnerable community identification using hate speech detection on social media. Inf Process Manage. https://doi.org/10.1016/j.ipm.2019.102087
52. Moule RK et al (2017) Technology and conflict: group processes and collective violence in the Internet era. Crime Law Soc Change 68(1–2):47–73
53. Poletti C, Michieli M (2018) Smart cities, social media platforms and security: online content regulation as a site of controversy and conflict. City Territ Archit 5(1):20. https://doi.org/10.1186/s40410-018-0096-2
54. Rajadesingan A, et al. Sarcasm detection on twitter: a behavioral modeling approach. In: Proceedings of the eighth ACM international conference on web search and data mining; 2015. P. 97–106
55. Saleem HM, et al. A Web of hate: tackling hateful speech in online social spaces. arXiv:1709.10159[cs]. 2017

56. Sahlgren M, et al. Learning representations for detecting abusive language. In: Proceedings of the 2nd workshop on abusive language online (ALW2), Brussels; 2018. P. 115–23
57. Gao L, Huang R. Detecting online hate speech using context aware models. arXiv preprint arXiv:1710.07395. 2017
58. Zhang Z et al (2018) Detecting hate speech on twitter using a convolution-gru based deep neural network. Eur Semant Web Conf 2018:745–760
59. Park JH, Fung P. One-step and two-step classification for abusive language detection on twitter. arXiv preprint arXiv:1706.01206. 2017
60. Unsvag EF, Gambäck B. The effects of user features on twitter hate speech detection. In: Proceedings of the 2nd workshop on abusive language online (ALW2) (2018), 75–85
61. Jafarpour B, Matwin S. Boosting text classification performance on sexist tweets by text augmentation and text generation using a combination of knowledge graphs. In: Proceedings of the 2nd workshop on abusive language online (ALW2); 2018. P. 107–14
62. Founta A-M, et al. A unified deep learning architecture for abuse detection. 2018
63. Qian J, et al. Leveraging intra-user and inter-user representation learning for automated hate speech detection. In: Proceedings of the 2018 conference of the north american chapter of the association for computational linguistics: human language technologies, volume 2 (Short Papers), New Orleans; 2018. P. 118–23
64. Chandrasekharan E, et al. The bag of communities: identifying abusive behavior online with preexisting internet data. In: Proceedings of the 2017 CHI conference on human factors in computing systems, New York; 2017. P. 3175–87
65. Waseem Z, Hovy D (2016) Hateful symbols or hateful people? predictive features for hate speech detection on twitter. Proc NAACL Stud Res Workshop 2016:88–93
66. Davidson T, et al. Racial bias in hate speech and abusive language detection datasets. In: Proceedings of the third workshop on abusive language online, Florence; 2019. P. 25–35
67. ElSherief M, et al. Hate Lingo: A Target-based linguistic analysis of hate speech in social media. In: The proceedings of the twelfth international AAAI conference on web and social media, Palo Alto; 2018
68. Saha P, et al. Hateminers: detecting hate speech against women. arXiv preprint arXiv:1812.06700. 2018
69. Qian J, et al. A benchmark dataset for learning to intervene in online hate speech. arXiv:1909.04251[cs]. 2019
70. Ross B, et al. Measuring the reliability of hate speech annotations: the case of the European refugee crisis. arXiv preprint arXiv:1701.08118. 2017
71. de Gibert O, et al. Hate speech dataset from a white supremacy forum. In: Proceedings of the 2nd workshop on abusive language online (ALW2), Brussels; 2018. P. 11–20
72. Salminen J, et al. Online hate interpretation varies by country, but more by individual: a statistical analysis using crowd sourced ratings. In: Proceedings of the fifth international conference on social networks analysis, management and security (SNAMS-2018), Valencia; 2018
73. Almerekhi H, et al. Detecting toxicity triggers in online discussions. In: The proceedings of the 30th ACM conference on hypertext and social media (HT'19), Hof; 2019
74. Fortuna PCT (2017) Automatic detection of hate speech in text: an overview of the topic and dataset annotation with hierarchical classes. Faculdade De Engenharia Da Universidade Do Porto, Porto
75. Salminen J, et al. Online hate ratings vary by extremes: a statistical analysis. In: Proceedings of the 2019 conference on human information interaction and retrieval, New York; 2019. P. 213–217
76. Gröndahl T, et al. All you need is"Love": evading hate-speech detection. arXiv preprint arXiv:1808.09115. 2018
77. Gunasekara I, Nejadgholi I. A review of standard text classification practices for multi-label toxicity identification of online content. In: Proceedings of the 2nd workshop on abusive language online (ALW2); 2018. P. 21–5
78. Xiang G, et al. Detecting offensive tweets via topical feature discovery over a large scale twitter corpus. In: Proceedings of the 21st ACM international conference on Information and knowledge management; 2012. P. 1980–4
79. Wang S, Manning CD. Baselines and bigrams: Simple, good sentiment and topic classification. In: Proceedings of the 50th annual meeting of the association for computational linguistics: Short papers-volume 2; 2012. P. 90–4
80. Dinakar K, et al. Modeling the detection of textual cyberbullying. In: Fifth international AAAI conference on weblogs and social media; 2011
81. Chen Y, et al. Detecting offensive language in social media to protect adolescent online safety. In: 2012 international conference on privacy, security, risk and trust and 2012 international conference on social computing; 2012. P. 71–80
82. Kwok I, Wang Y. Locate the hate: Detecting tweets against blacks. In: Twenty-seventh AAAI conference on artificial intelligence; 2013
83. Xu J-M, et al. Learning from bullying traces in social media. In: Proceedings of the 2012 conference of the North American chapter of the association for computational linguistics: Human language technologies; 2012. P. 656–66
84. Dadvar M et al (2013) Improving cyberbullying detection with user context. Eur Conf Inf Retriev 2013:693–696
85. Chen T, Guestrin C. Xgboost: A scalable tree boosting system. In: Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining; 2016. P. 785–94
86. Li Y, Yuan Y (2017) Convergence analysis of two-layer neural networks with relu activation. Adv Neural Inf Process Syst 2017:597–607
87. Kingma DP, Ba J. Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980. 2014
88. Mikolov T, et al. Distributed representations of words and phrases and their compositionality. In: Advances in neural information processing systems 26. C.J.C. Burges et al., eds. Curran Associates, Inc. p. 3111–9
89. Le Q, Mikolov T. Distributed representations of sentences and documents. In: Proceedings of the 31st international conference on machine learning (ICML-14); 2014. P. 1188–96
90. Kshirsagar R, et al. Predictive embeddings for hate speech detection on twitter. arXiv preprint arXiv:1809.10644. 2018
91. Devlin J, et al. Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805. 2018

92. Aggarwal P, et al. LTL-UDE at SemEval-2019 Task 6: BERT and two-vote classification for categorizing offensiveness. In: Proceedings of the 13th international workshop on semantic evaluation, Minneapolis; 2019. P. 678–82
93. Liu P, et al. NULI at SemEval-2019 Task 6: transfer learning for offensive language detection using bidirectional transformers. In: Proceedings of the 13th international workshop on semantic evaluation, Minneapolis; 2019. P. 87–91
94. Nikolov A, Radivchev V. SemEval-2019 Task 6: offensive tweet classification with BERT and ensembles. In: Proceedings of the 13th international workshop on semantic evaluation, Minneapolis; 2019. P. 691–5
95. Zampieri M, et al. SemEval-2019 Task 6: identifying and categorizing offensive language in social media (OffensEval). arXiv:1903.08983[cs]. 2019
96. Zhu J, et al. UM-IU@LING at SemEval-2019 Task 6: Identifying Offensive Tweets Using BERT and SVMs. arXiv :1904.03450[cs]. 2019
97. Tausczik YR, Pennebaker JW (2010) The psychological meaning of words: LIWC and computerized text analysis methods. J Lang Soc Psychol 29(1):24–54
98. McCarthy PM, Boonthum-Denecke C (eds) (2012) Applied natural language processing: identification, investigation and resolution. Hershey, IGI Global
99. Young HP (1985) Monotonic solutions of cooperative games. Int J Game Theory 14(2):65–72
100. Aroyo L, et al. Crowdsourcing subjective tasks: the case study of understanding toxicity in online discussions. In: Companion proceedings of the 2019 world wide web conference, San Francisco; 2019. P. 1100–5
101. Tian H, Chen P-Y (2019) "I'm in the center of the vortex": the affective chain of social media trolling. Proc Assoc Inf Sci Technol 56(1):778–779. https://doi.org/10.1002/pra2.173
102. Berglind T, et al. Levels of hate in online environments. In: Proceedings of the IEEE/ACM international conference on advances in social networks analysis and mining (ASONAM), Vancouver; 2019
103. MacAvaney S et al (2019) Hate speech detection: challenges and solutions. PLoS ONE 14(8):e0221152. https://doi.org/10.1371/journal.pone.0221152
104. Ullmann S, Tomalin M (2019) Quarantining online hate speech: technical and ethical perspectives. Ethics Inf Technol. https://doi.org/10.1007/s10676-019-09516-z
105. Holzinger A (2016) Interactive machine learning for health informatics: when do we need the human-in-the-loop? Brain Inf 3(2):119–131
106. Mariconti E, et al. You Know What to Do. In: Proceedings of the ACM on human–computer interaction. 2019
107. Yang Z, et al. XLNet: generalized autoregressive pretraining for language understanding. arXiv:1906.08237[cs]. 2019
108. Murray A (2019) Information technology law: the law and society. Oxford University Press, Oxford
109. Walker S (1994) Hate speech: the history of an American controversy. Nebraska Press, Lincoln
110. Sap M, et al. The risk of racial bias in hate speech detection. In: Proceedings of the 57th annual meeting of the association for computational linguistics, Florence; 2019. P. 1668–78
111. Chatzakou D, et al. Hate is not binary: studying abusive behavior of #GamerGate on Twitter. In: Proceedings of the 28th ACM conference on hypertext and social media, New York; 2017. P. 65–74
112. Chatzakou D, et al. Mean birds: detecting aggression and bullying on twitter. In: Proceedings of the 2017 ACM on Web Science Conference, New York; 2017. P. 13–22
113. Agarwal S, Sureka A. A focused crawler for mining hate and extremism promoting videos on YouTube. In: Proceedings of the 25th ACM conference on hypertext and social media, New York; 2014. P. 294–6

Publisher's Note