Automated Hate Speech Detection and the Problem of Offensive Language*

Thomas Davidson,[1] **Dana Warmsley,**[2] **Michael Macy,**[1,3] **Ingmar Weber**[4]
[1]Department of Sociology, Cornell University, Ithaca, NY, USA
[2]Department of Applied Mathematics, Cornell University, Ithaca, NY, USA
[3]Department of Information Science, Cornell University, Ithaca, NY, USA
[4]Qatar Computing Research Institute, HBKU, Doha, Qatar
{trd54, dw457, mwmacy}@cornell.edu, iweber@hbku.edu.qa

Abstract

A key challenge for automatic hate-speech detection on social media is the separation of hate speech from other instances of offensive language. Lexical detection methods tend to have low precision because they classify all messages containing particular terms as hate speech and previous work using supervised learning has failed to distinguish between the two categories. We used a crowd-sourced hate speech lexicon to collect tweets containing hate speech keywords. We use crowd-sourcing to label a sample of these tweets into three categories: those containing hate speech, only offensive language, and those with neither. We train a multi-class classifier to distinguish between these different categories. Close analysis of the predictions and the errors shows when we can reliably separate hate speech from other offensive language and when this differentiation is more difficult. We find that racist and homophobic tweets are more likely to be classified as hate speech but that sexist tweets are generally classified as offensive. Tweets without explicit hate keywords are also more difficult to classify.

Introduction

What constitutes hate speech and when does it differ from offensive language? No formal definition exists but there is a consensus that it is speech that targets disadvantaged social groups in a manner that is potentially harmful to them (Jacobs and Potter 2000; Walker 1994). In the United States, hate speech is protected under the free speech provisions of the First Amendment, but it has been extensively debated in the legal sphere and with regards to speech codes on college campuses. In many countries, including the United Kingdom, Canada, and France, there are laws prohibiting hate speech, which tends to be defined as speech that targets minority groups in a way that could promote violence or social disorder. People convicted of using hate speech can often face large fines and even imprisonment. These laws extend to the internet and social media, leading many sites to create their own provisions against hate speech. Both Facebook and Twitter have responded to criticism for not doing enough to prevent hate speech on their sites by instituting policies to prohibit the use of their platforms for attacks on people based on characteristics like race, ethnicity, gender, and sexual orientation, or threats of violence towards others.[1]

Drawing upon these definitions, we define hate speech as *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*. In extreme cases this may also be language that threatens or incites violence, but limiting our definition only to such cases would exclude a large proportion of hate speech. Importantly, our definition does not include all instances of offensive language because people often use terms that are highly offensive to certain groups but in a qualitatively different manner. For example some African Americans often use the term *n*gga*[2] in everyday language online (Warner and Hirschberg 2012), people use terms like *h*e* and *b*tch* when quoting rap lyrics, and teenagers use homophobic slurs like *f*g* as they play video games. Such language is prevalent on social media (Wang et al. 2014), making this boundary condition crucial for any usable hate speech detection system .

Previous work on hate speech detection has identified this problem but many studies still tend to conflate hate speech and offensive language. In this paper we label tweets into three categories: hate speech, offensive language, or neither. We train a model to differentiate between these categories and then analyze the results in order to better understand how we can distinguish between them. Our results show that fine-grained labels can help in the task of hate speech detection and highlights some of the key challenges to accurate classification. We conclude that future work must better account for context and the heterogeneity in hate speech usage.

Related Work

Bag-of-words approaches tend to have high recall but lead to high rates of false positives since the presence of offensive words can lead to the misclassification of tweets as hate speech (Kwok and Wang 2013; Burnap and Williams

*This is a preprint of a short paper accepted at ICWSM 2017. Please cite that version instead.

[1]Facebook's policy can be found here: `www.facebook.com/communitystandards#hate-speech`. Twitter's policy can be found here: `support.twitter.com/articles/20175050`.

[2]Where present, the "*" has been inserted by us and was not part of the original text. All tweets quoted have been modified slightly to protect user's identities while retaining their original meaning.

2015). Focusing on anti-black racism, Kwok and Wang find that 86% of the time the reason a tweet was categorized as racist was because it contained offensive words. Given the relatively high prevalence of offensive language and "curse words" on social media this makes hate speech detection particularly challenging (Wang et al. 2014). The difference between hate speech and other offensive language is often based upon subtle linguistic distinctions, for example tweets containing the word *n*gger* are more likely to be labeled as hate speech than *n*gga* (Kwok and Wang 2013). Many can be ambiguous, for example the word *gay* can be used both pejoratively and in other contexts unrelated to hate speech (Wang et al. 2014).

Syntactic features have been leveraged to better identify the targets and intensity of hate speech, for example sentences where a relevant noun and verb occur (e.g. *kill* and *Jews*) (Gitari et al. 2015), the POS trigram "DT jewish NN" (Warner and Hirschberg 2012), and the syntactic structure I <intensity > <user intent > <hate target >, e.g. "I f*cking hate white people" (Silva et al. 2016).

Other supervised approaches to hate speech classification have unfortunately conflated hate speech with offensive language, making it difficult to ascertain the extent to which they are really identifying hate speech (Burnap and Williams 2015; Waseem and Hovy 2016). Neural language models show promise in the task but existing work has used training data has a similarly broad definition of hate speech (Djuric et al. 2015). Non-linguistic features like the gender or ethnicity of the author can help improve hate speech classification but this information is often unavailable or unreliable on social media (Waseem and Hovy 2016).

Data

We begin with a hate speech lexicon containing words and phrases identified by internet users as hate speech, compiled by *Hatebase.org*. Using the Twitter API we searched for tweets containing terms from the lexicon, resulting in a sample of tweets from 33,458 Twitter users. We extracted the time-line for each user, resulting in a set of 85.4 million tweets. From this corpus we then took a random sample of 25k tweets containing terms from the lexicon and had them manually coded by CrowdFlower (CF) workers. Workers were asked to label each tweet as one of three categories: hate speech, offensive but not hate speech, or neither offensive nor hate speech. They were provided with our definition along with a paragraph explaining it in further detail. Users were asked to think not just about the words appearing in a given tweet but about the context in which they were used. They were instructed that the presence of a particular word, however offensive, did not necessarily indicate a tweet is hate speech. Each tweet was coded by three or more people. The intercoder-agreement score provided by CF is 92%. We use the majority decision for each tweet to assign a label. Some tweets were not assigned labels as there was no majority class. This results in a sample of 24,802 labeled tweets.

Only 5% of tweets were coded as hate speech by the majority of coders and only 1.3% were coded unanimously, demonstrating the imprecision of the Hatebase lexicon. This is much lower than a comparable study using Twitter, where 11.6% of tweets were flagged as hate speech (Burnap and Williams 2015), likely because we use a stricter criteria for hate speech. The majority of the tweets were considered to be offensive language (76% at 2/3, 53% at 3/3) and the remainder were considered to be non-offensive (16.6% at 2/3, 11.8% at 3/3). We then constructed features from these tweets and used them to train a classifier.

Features

We lowercased each tweet and stemmed it using the Porter stemmer,[3] then create bigram, unigram, and trigram features, each weighted by its TF-IDF. To capture information about the syntactic structure we use `NLTK` (Bird, Loper, and Klein 2009) to construct Penn Part-of-Speech (POS) tag unigrams, bigrams, and trigrams. To capture the quality of each tweet we use modified Flesch-Kincaid Grade Level and Flesch Reading Ease scores, where the number of sentences is fixed at one. We also use a sentiment lexicon designed for social media to assign sentiment scores to each tweet (Hutto and Gilbert 2014). We also include binary and count indicators for hashtags, mentions, retweets, and URLs, as well as features for the number of characters, words, and syllables in each tweet.

Model

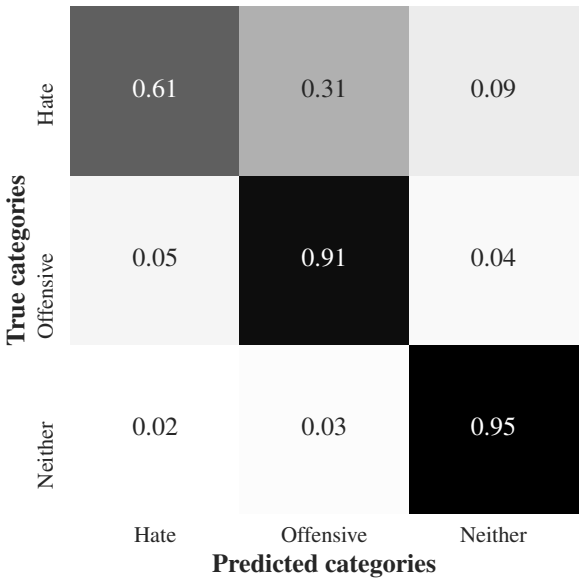
We first use a logistic regression with L1 regularization to reduce the dimensionality of the data. We then test a variety of models that have been used in prior work: logistic regression, naïve Bayes, decision trees, random forests, and linear SVMs. We tested each model using 5-fold cross validation, holding out 10% of the sample for evaluation to help prevent over-fitting. After using a grid-search to iterate over the models and parameters we find that the Logistic Regression and Linear SVM tended to perform significantly better than other models. We decided to use a logistic regression with L2 regularization for the final model as it more readily allows us to examine the predicted probabilities of class membership and has performed well in previous papers (Burnap and Williams 2015; Waseem and Hovy 2016). We trained the final model using the entire dataset and used it to predict the label for each tweet. We use a one-versus-rest framework where a separate classifier is trained for each class and the class label with the highest predicted probability across all classifiers is assigned to each tweet. All modeling was performing using `scikit-learn` (Pedregosa and others 2011).

Results

The best performing model has an overall precision 0.91, recall of 0.90, and F1 score of 0.90. Looking at *Figure 1*, however, we see that almost 40% of hate speech is misclassified: the precision and recall scores for the hate class are 0.44 and 0.61 respectively. Most of the misclassification occurs in the upper triangle of this matrix, suggesting that the

[3]We verified that the stemmer did not remove important information by reducing key terms to the same stem, e.g. *f*gs* and *f*ggots* stem to *f*g* and *f*ggot*.

Figure 1: **True versus predicted categories**



model is biased towards classifying tweets as less hateful or offensive than the human coders. Far fewer tweets are classified as more offensive or hateful than their true category; approximately 5% of offensive and 2% of innocuous tweets have been erroneously classified as hate speech. To explore why these tweets have been misclassified we now look more closely at the tweets and their predicted classes.

Tweets with the highest predicted probabilities of being hate speech tend to contain multiple racial or homophobic slurs, e.g. *@JuanYeez shut yo beaner ass up sp*c and hop your f*ggot ass back across the border little n*gga* and *RT @eBeZa: Stupid f*cking n*gger LeBron. You flipping jungle bunny monkey f*ggot*. Other tweets tend to be correctly identified as hate when they contained strongly racist or homophobic terms like *n*gger* and *f*ggot*. Interestingly, we also find cases where people use hate speech to respond to other hate speakers, such as this tweet where someone uses a homophobic slur to criticize someone else's racism: *@MrMoonfrog @RacistNegro86 f*ck you, stupid ass coward b*tch f*ggot racist piece of sh*t*.

Turning to true hate speech classified as offensive it appears that tweets with the highest predicted probability of being offensive are genuinely less hateful and were perhaps mislabeled, for example *When you realize how curiosity is a b*tch #CuriosityKilledMe* may have been erroneously coded as hate speech if people thought that *curiosity* was a person, and *Why no boycott of racist "redskins"? #Redskins #ChangeTheName* contains a slur but is actually against racism. It is likely that coders skimmed these tweets too quickly, picking out words or phrases that appeared to be hateful without considering the context. Turning to borderline cases, where the probability of being offensive is marginally higher than hate speech, it appears that the majority are hate speech, both directed towards other Twitter users, *@MDreyfus @NatFascist88 Sh*t your ass your moms p*ssy u Jew b*stard. Ur times coming. Heil Hitler!* and general hateful statements like *My advice of the day: If your a tranny...go f*ck your self!*. These tweets fit our definition of hate speech but were likely misclassified because they do not contain any of the terms most strongly associated with hate speech. Finally, the hateful tweets incorrectly labeled as neither tend not to contain hate or curse words, for example *If some one isn't an Anglo-Saxon Protestant, they have no right to be alive in the US. None at all, they are foreign filth* contains a negative term, *filth* but no slur against a particular group. We also see that rarer types of hate speech, for example this anti-Chinese statement *Every slant in #LA should be deported. Those scum have no right to be here. Chinatown should be bulldozed*, are incorrectly classified. While the classifier performs well at prevalent forms of hate speech, particularly anti-black racism and homophobia, but is less reliable at detecting types of hate speech that occur infrequently, a problem noted by Nobata et al. (2016).

A key flaw in much previous work is that offensive language is mislabeled as hate speech due to an overly broad definition. Our multi-class framework allows us to minimize these errors; only 5% of our true offensive language was labeled as hate. The tweets correctly labeled as offensive tend to contain curse words and often sexist language, e.g. *Why you worried bout that other h*e? Cuz that other h*e aint worried bout another h*e* and *I knew Kendrick Lamar was onto something when he said "I call a b*tch a b*tch, a h*e a h*e, a woman a woman"*. Many of these tweets contain sexist terms like *b*tch*, *p*ssy*, and *h*e*. Human coders appear to consider racists or homophobic terms to be hateful but consider words that are sexist and derogatory towards women to be only offensive, consistent prior findings (Waseem and Hovy 2016).

Looking at the tweets misclassified as hate speech we see that many contain multiple slurs, e.g. *@SmogBaby: These h*es be lyin to all of us n*ggas* and *My n*gga mister meaner just hope back in the b*tch*. While these tweets contain terms that can be considered racist and sexist it is apparent than many Twitter users use this type of language in their everyday communications. When they do contain racist language they tend to contain the term *n*gga* rather than *n*gger*, in line with the findings of Kwok and Wang (2013). We also found a few recurring phrases such as *these h*es ain't loyal* that were actually lyrics from rap songs that users were quoting. Classification of such tweets as hate speech leads us to overestimate the prevalence of the phenomenon. While our model still misclassifies some offensive language as hate speech we are able to avoid the vast majority of these errors by differentiating between the two.

Finally, turning to the neither class, we see that tweets with the highest predicted probability of belonging to this class all appear to be innocuous and were included in the sample because they contained terms included in the Hatebase lexicon such as *charlie* and *bird* that are generally not used in a hateful manner. Tweets with overall positive sen-

timent and higher readability scores are more likely to belong to this class. The tweets in this category that have been misclassified as hate or offensive tend to mention race, sexuality, and other social categories that are targeted by hate speakers. Most appear to be misclassifications appear to be caused by on the presence of potentially offensive language, for example *He's a damn good actor. As a gay man it's awesome to see an openly queer actor given the lead role for a major film* contains the potentially the offensive terms *gay* and *queer* but uses them in a positive sense. This problem has been encountered in previous research (Warner and Hirschberg 2012) and illustrates the importance of taking context into account. We also found a small number of cases where the coders appear to have missed hate speech that was correctly identified by our model, e.g. *@mayormcgunn @SenFeinstein White people need those weapons to defend themselves from the subhuman trash your sort unleashes on us*. This finding is consistent with previous work that has found amateur coders to often be unreliable at identifying abusive content (Nobata et al. 2016; Waseem 2016).

Conclusions

If we conflate hate speech and offensive language then we erroneously consider many people to be hate speakers (errors in the lower triangle of Figure 1) and fail differentiate between commonplace offensive language and serious hate speech (errors in the upper triangle of Figure 1). Given the legal and moral implications of hate speech it is important that we are able to accurately distinguish between the two. Lexical methods are effective ways to identify potentially offensive terms but are inaccurate at identifying hate speech; only a small percentage of tweets flagged by the Hatebase lexicon were considered hate speech by human coders.[4] While automated classification methods can achieve relatively high accuracy at differentiating between these different classes, close analysis of the results shows that the presence or absence of particular offensive or hateful terms can both help and hinder accurate classification.

Consistent with previous work, we find that certain terms are particularly useful for distinguishing between hate speech and offensive language. While *f*g*, *b*tch*, and *n*gga* are used in both hate speech and offensive language, the terms *f*ggot* and *n*gger* are generally associated with hate speech. Many of the tweets considered most hateful contain multiple racial and homophobic slurs. While this allows us to easily identify some of the more egregious instances of hate speech it means that we are more likely to misclassify hate speech if it doesn't contain any curse words or offensive terms. To more accurately classify such cases we should find sources of training data that are hateful without necessarily using particular keywords or offensive language.

Our results also illustrate how hate speech can be used in different ways: it can be directly send to a person or group of people targeted, it can be espoused to nobody in particular, and it can be used in conversation between people. Future work should distinguish between these different uses and look more closely at the social contexts and conversations in which hate speech occurs. We must also study more closely the people who use hate speech, focusing both on their individual characteristics and motivations and on the social structures they are embedded in.

Hate speech is a difficult phenomenon to define and is not monolithic. Our classifications of hate speech tend to reflect our own subjective biases. People identify racist and homophobic slurs as hateful but tend to see sexist language as merely offensive. While our results show that people perform well at identifying some of the more egregious instances of hate speech, particularly anti-black racism and homophobia, it is important that we are cognizant of the social biases that enter into our algorithms and future work should aim to identify and correct these biases.

[4]If a lexicon must be used we propose that a smaller lexicon with higher precision is preferable to a larger lexicon with higher recall. We have made a more restricted version of the Hatebase lexicon available here: `https://github.com/t-davidson/hate-speech-and-offensive-language`.

References

Bird, S.; Loper, E.; and Klein, E. 2009. *Natural Language Processing with Python*. O'Reilly Media Inc.

Burnap, P., and Williams, M. L. 2015. Cyber hate speech on twitter: An application of machine classification and statistical modeling for policy and decision making. *Policy & Internet* 7(2):223–242.

Djuric, N.; Zhou, J.; Morris, R.; Grbovic, M.; Radosavljevic, V.; and Bhamidipati, N. 2015. Hate speech detection with comment embeddings. In *WWW*, 29–30.

Gitari, N. D.; Zuping, Z.; Damien, H.; and Long, J. 2015. A lexicon-based approach for hate speech detection. *International Journal of Multimedia and Ubiquitous Engineering* 10:215–230.

Hutto, C. J., and Gilbert, E. 2014. VADER: A parsimonious rule-based model for sentiment analysis of social media text. In *ICWSM*.

Jacobs, J. B., and Potter, K. 2000. *Hate crimes: Criminal Law and Identity Politics*. Oxford University Press.

Kwok, I., and Wang, Y. 2013. Locate the hate: Detecting tweets against blacks. In *AAAI*.

Nobata, C.; Tetreault, J.; Thomas, A.; Mehdad, Y.; and Chang, Y. 2016. Abusive language detection in online user content. In *WWW*, 145–153.

Pedregosa, F., et al. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research* 12:2825–2830.

Silva, L. A.; Mondal, M.; Correa, D.; Benevenuto, F.; and Weber, I. 2016. Analyzing the targets of hate in online social media. In *ICWSM*, 687–690.

Walker, S. 1994. *Hate Speech: The History of an American Controversy*. U of Nebraska Press.

Wang, W.; Chen, L.; Thirunarayan, K.; and Sheth, A. P. 2014. Cursing in english on twitter. In *CSCW*, 415–425.

Warner, W., and Hirschberg, J. 2012. Detecting hate speech on the world wide web. In *LSM*, 19–26.

Waseem, Z., and Hovy, D. 2016. Hateful symbols or hateful people? predictive features for hate speech detection on twitter. In *SRW@HLT-NAACL*, 88–93.

Waseem, Z. 2016. 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 CSS*, 138–142.

A benchmark for toxic comment classification on Civil Comments dataset

Corentin Duchêne, Henri Jamet, Pierre Guillaume, Réda Dehak

firstname.lastname@epita.fr
EPITA Speaker and Language Recognition Group (ESLR),
Laboratoire de Recherche de l'EPITA (LRE), France

Abstract. Toxic comment detection on social media has proven to be essential for content moderation. This paper compares a wide set of different models on a highly skewed multi-label hate speech dataset. We consider inference time and several metrics to measure performance and bias in our comparison. We show that all BERTs have similar performance regardless of the size, optimizations or language used to pre-train the models. RNNs are much faster at inference than any of the BERT. BiLSTM remains a good compromise between performance and inference time. RoBERTa with Focal Loss offers the best performance on biases and AUROC. However, DistilBERT combines both good AUROC and a low inference time. All models are affected by the bias of associating identities. BERT, RNN, and XLNet are less sensitive than the CNN and Compact Convolutional Transformers.

1 Introduction

Toxic comment detection on social media has proven to be essential for content moderation. According to the French Minister of Education, 18% of French students were victims of harassment on social networks in 2021. At the same time, the number of posts on these platforms has been increasing. In 12 years, the number of tweets per day has increased tenfold to reach 500 million today[1].

This shows that the rapid and targeted detection of toxic comments on social networks became a crucial issue in ensuring the cohesion of society. Therefore, this can only be done by automating online moderation.

Nowadays, the types of models performing well on text classification and representing state-of-the-art are transformer-based models (Vaswani et al., 2017) such as BERT (Devlin et al., 2019). Liu et al. (2019) refined a pre-trained BERT model for identifying offensive language, automatically categorizing hate types, and identifying the target of the comment. Swamy et al. (2019) used BERT for training and testing cross-data sets, Isaksen and Gambäck (2020) separately refined BERT on several datasets for hate speech and offensive language detection.

[1] Twitter usage statistics - Internet Live Stats

In this study, we compare state-of-the-art models in natural language processing, such as BERT, and in vision applied to text, such as ResNet and Vision Transformers. To our best knowledge, we did not find in the state-of-the-art a detailed comparison of all these models on a wide range of metrics using the same training conditions and same training and testing datasets. As never before, the same methodology and dataset are used throughout our analysis to focus on performance, bias measurement, and inference time. We have tuned each of our models to achieve the best performance. This work's result should help determine which model can be used in practice.

Our comparison was performed using the same training and testing datasets extracted from the Civil Comments 2019[2]. This dataset is a multi-label dataset with imbalanced classes provided by Jigsaw/Conversation AI. For this dataset, we know the targeted identity for some comments, so that we can evaluate the biases during classification.

The rest of the paper is organized as follows: Section 2 describes the dataset, and the models used in the comparison. The experiment protocol and the results' analysis are presented in sections 3 and 4. Finally, section 5 concludes the paper.

2 Methodology

2.1 Dataset

In 2017, the comment-hosting platform Civil Comments closed. The company has made its 1.8 million comments public to support research to understand and improve civility detection in online conversations. The Jigsaw team supports this action; each comment was shown to 10 annotators and asked them to "Rate the toxicity of the comment". To ensure the accuracy of the ratings, some comments were seen by more than 100 annotators. For all comments, the value obtained at the end for each class is the fraction of positive annotations over the number of annotators. All comments were classified into seven categories: `toxicity`, `severe_toxicity`, `obscene`, `threat`, `insult`, `indentity_attack`, and `sexual_explicit`.

Category	**Identity Options**
Gender	Male, Female, Transgender, Other gender
Sexual Orientation	Heterosexual, Homosexual, Bisexual, Other sexual orientation
Religion	Christian, Jewish, Muslim, Hindu, Buddhist, Atheist, Other religion
Race or ethnicity	Black, White, Latino, Other race or ethnicity
Disability	Physical disability, Intellectual or learning disability. Psychiatric disability or mental illness, Other disability

TAB. 1 – *List of identity options presented to the annotators.*

[2]Jigsaw unintended bias in toxicity classification Kaggle

Subgroup	Count	Percent Toxic
all comments	1 999 516	7.99%
male	48 870	15.05%
female	58 584	13.66%
transgender	2 759	21.13%
heterosexual	1 432	22.56%
homosexual	12 062	28.28%

TAB. 2 – *Percentage of comments labeled toxic for a selection of identities.*

In addition, a subset of 450,000 examples from this dataset was tagged with identity (Table 1) using a list of questions, such as "What genders are referenced in the comment?" or "What races or ethnicities are referenced in the comment?". Again, the score obtained for each identity class is the fraction of annotators who mentioned the identity out of the number of evaluators. We can see in Table 2 that there is an imbalance in toxicity percentage annotation between different identities.

2.2 Preprocessing

The comments are labeled with the probabilities of belonging to a class (for all toxicity and identity classes). To determine whether a comment is considered positive or negative for a class, we applied a threshold: if the probability is greater than 0.5; we assume that the comment is positive for that class, otherwise negative.

We notice that the classes are highly unbalanced. The label `severe_toxicity` is rarely activated on the whole dataset, as shown in Table 3. For this reason, this class has been removed from the classes to be predicted to limit the number of classes to six.

Hate subtype	Count
toxicity	159 782
severe_toxicity	13
obscene	10 671
sexual_explicit	5 127
identity_attack	14 761
insult	118 079
threat	4 725

TAB. 3 – *Count of comments for each subtype of hate speech.*

The following transformations are applied to each comment:

- Remove HTML tags
- Remove URL
- Remove diacritics
- transform to lowercase

- Remove white space
- Remove NA or empty

The dataset available on Kaggle[2] already provided a split into train and test subsets. It is assumed that the distributions of labels and subgroups between the two subsets are similar but not exact.

To deal with the problem of unbalanced dataset, during the training step, we re-balance the toxicity classes. To do this, we will apply a negative down-sampling: we keep only 10% of the randomly chosen examples without toxicity (all 6 toxicity classes not enabled), and we keep all the examples with at least one of the 6 classes enabled. In fine, there are as many examples with all the negative classes as there are examples with at least one positive class. In total, the size of the training set is 310 000 examples. It is important to note that no re-balancing is done on the test subset.

2.3 Models

Most of the trained transformers are based on BERT, it stands for Bidirectional Encoder Representations from Transformers, Google developed it in 2018. It is a Transformer-based model that only uses the encoder part of the Transformer. BERT model can also be used for classification. It uses a specific token `<CLS>` in the beginning of each sequence for classification purposes. In our comparison, we assume BERT as our baseline model.

Despite the excellent results in different benchmarks, this is a model that has some limitations. Since the release of BERT, different models were proposed to address some BERT limitations. For this reason, we will investigate the performance of recent transformer language models: DistilBERT (Sanh et al., 2019), AlBERT (Lan et al., 2019), RoBERTa (Zhuang et al., 2021), XLM RoBERTa (Conneau et al., 2020), BERTweet (Nguyen et al., 2020), HateBERT (Caselli et al., 2021), XLNet (Yang et al., 2019) and Compact Convolutional Transformer (CCT) (Hassani et al., 2021).

DistilBERT was proposed by Sanh et al. (2019). It is a distilled (Hinton et al., 2014) version of the BERT model. The new model has 40% less parameters, runs 60% faster while preserving over 95% of BERT's performances.

AlBERT (Lan et al., 2019), which stands for "A Lite BERT", was made available in an open source version by Google in 2019. The model was built with the original BERT structure, but designed to drastically reduce parameters (by 89%) using sharing parameters across the hidden layers of the network, and factorizing the embedding layer. All of this was accomplished with an accuracy reduction of 82.3% to 80.1% on average over a list of datasets.

RoBERTa (Zhuang et al., 2021) is a modification of BERT model proposed by Facebook AI in 2019. To improve end-task performance, Roberta uses a byte-level Byte-Pair-Encoding (Sennrich et al., 2016) as a tokenizer. Hence, the tokenizer contains more than 50k words, which increases the embedding layer size and the number of learning parameters. Regarding prior training, Roberta has been trained on Masked Language Modeling (MLM) and Causal Language Modeling (CLM). In causal language modeling, the model tries to predict masked

token with only the left or right tokens in the sentence, which makes the prediction unidirectional.

XLM RoBERTa (Conneau et al., 2020) is a multilingual version of RoBERTa. It is pre-trained on 2.5 TB of filtered CommonCrawl data containing 100 languages.

BERTweet (Nguyen et al., 2020) is a BERT-based model trained on a huge English tweet corpus proposed by Nvidia. It was trained using the Roberta procedure on language modeling task. The corpus used for the training is about 820 millions (80Go) of English tweets. In order to train the model, huge capacity resources were needed: 8 v100 GPUs with 32 GB each. BERTweet has showed to outperform Roberta base model on the following tweets task: Part-of-speech tagging, Named-entity recognition and text classification.

HateBERT (Caselli et al., 2021) is a model published in Association for Computational Linguistics conference. It uses a pretrained BERT base model. This model has been fine-tuned for a language modeling task on specific social network dataset RAL-E (Reddit Abusive Language English dataset). The dataset is made of 1 492 740 different sentences from Reddit and contains hate speech, offensive and abusive phrases. The model has also been fine-tuned on 3 different datasets: OfensEval, AbusEval and HatEval beating the state of the art on these 3 datasets.

XLNet (Yang et al., 2019) is a large bidirectional transformer that uses improved training methodology, larger training dataset, and more computational power. XLNet outperformed BERT on 20 tasks, such as question answering, natural language inference, sentiment analysis, etc.

For all these models, we concatenate the output of the last 4 layers in one large features vector and stacked two dense layers to get a vector of size 6 which corresponds to the 6 toxicity classes to be predicted. Pre-trained models were used, and the models weights were unfrozen during training. Multiple research have been done regarding features extraction in Transformers. Results presented in (Devlin et al., 2019) inspired our study and comparison. The paper shows that the concatenation of the four last layers from the encoder gives better results than using only the last layer.

Compact Convolutional Transformer (CCT) (Hassani et al., 2021) is a Transformer-based architecture for vision. The original paper shows that CCT can lead to good results on image and on text datasets with fewer parameters compared to Transformer based models. In previous research, some sought to use transformers on images: Vision Transformer (ViT) (Dosovitskiy et al., 2021). The main idea of these models is to use the advantages of Transformers on images to extract information that cannot be brought out by convolutions. Unlike ViT, CCT combines convolutions and Transformer attention layers. CCT first uses a convolution tokenization on the image, while Vision Transformer uses patch-based tokenization. This layer applies a certain number of convolutions that produce a set of maps that are reshaped (flatten) and directly used by an optional positional embedding layer. The embedding is then

fed to a series of Transformer encoder layers and pooled before being used by dense layers for classification. We have recovered this type of transformer for our study to use it again on text. Since CCT works only on images, we used Glove pre-trained embedding to represent the sentences from the dataset as images. We padded the sentences to a fixed length and concatenated each embedded word to form a matrix representing a one channel image. The training was done from scratch, and we used a pre-trained GloVe embedding enriched during the training.

Global Vectors for Word Representation (GloVe) (Pennington et al., 2014) is a model used to find word vectors. It uses a co-occurrence matrix to consider the global context of the words in the sentence. Semantic relationships between words can be extracted from the co-occurrence matrix.

To compare BERT-based models with other more traditional models, we trained a Bidirectional GRU and a Bidirectional LSTM from scratch. For each one, three layers of RNN and one layer of embedding unfreeze GloVe (Pennington et al., 2014) were used.

Several ResNet (He et al., 2016) with a depth of 44 and 56 were also trained from scratch. We used pre-trained GloVe embedding. In some sessions, we froze the embedding.

3 Experiments

3.1 Training

All models are trained over three epochs, with a batch size of 32 examples, except for CCT, where we limit ourselves to 8 per batch due to the lack of VRAM. We use the AdamW optimizer.

We use the positive weighted Binary Cross Entropy (pwBCE) as a loss function. This loss function adds weights on the positive samples to consider them as much as the negatives. For the RoBERTa model, we use three other loss functions which are the Binary Cross Entropy (BCE), Focal Loss (FL) (Lin et al., 2017), and positive weighted Focal Loss (pwFL). The FL reduces the loss attributed to well-ranked examples and focuses on examples with poorly ranked classes, usually due to class imbalance. pwFL corresponds to the same trick as for pwBCE applied to FL.

To measure the model's performance, we use similar metrics that were used during the kaggle[2]: Macro AUROC, Macro F1 and Micro F1 with a threshold of 0.5, Precision and Recall. To understand the model's complexity, we also measure the inference time per batch calculated on the test set. The average inference time per batch is computed from 6,000 batches during the inference phase on the test set.

As we can see in Table 4, the hate speech detection models could make biased predictions for particular identities who are already the target of such abuse. To measure such unintended model bias, we rely on the AUC-based metrics developed by Borkan et al. (2019). These include Subgroup AUC (Sub. AUC), Background Positive Subgroup Negative (BPSN) AUC, and Background Negative Subgroup Positive (BNSP) AUC.

The sub. AUC measures the AUROC for each identity using toxic and normal posts from the test set that mention the identity under consideration. A higher value means that a model is less likely to confuse the normal post that mentions the community with a toxic post that does not.

The BPSN AUC measures the AUROC for each identity, using normal posts that mention the identity and toxic posts that do not mention the identity under consideration. A higher value means that a model is less likely to confuse the normal post that mentions the community with a toxic post that does not.

The BNSP AUC measures the AUROC for each identity using toxic posts that mention the identity and normal posts that do not mention the identity under consideration from the test set. A higher value means that the model is less likely to confuse a toxic post that mentions the community with a normal post without one.

To combine these metrics across identities, we used the generalized mean (GM) or power mean with exponent p, which was already used by the Jigsaw/Conversation AI Team during a Kaggle competition[2]. So, we report the following three bias metrics for our comparison:

- **GMB-Subgroup-AUC** is the GM for the Subgroup AUC
- **GMB-BPSN-AUC** is the GM of the BPSN AUC
- **GMB-BNSP-AUC** is the GM of the BNSP AUC

We restrict the evaluation to the test set only. By having this restriction, we can evaluate models in terms of bias reduction. Only identities with more than 500 examples in the test dataset will be included in the evaluation calculation.

4 Results

			Performance					Bias		
Model type	Id	Model name	AUROC	Macro F1	Micro F1	Precision	Recall	GMB Sub.	GMB BPSN	GMB BNSP
BERT	0	AlBERT	0.9790	0.3463	0.4786	0.3247	0.9104	0.8674	0.8998	0.9513
	1	BERTweet	0.9816	0.3616	0.4928	0.3363	0.9216	0.8780	0.8945	0.9603
	2	DistilBERT	0.9804	0.3879	0.5115	0.3572	0.9001	0.8762	0.8740	**0.9644**
	3	HateBERT	0.9791	0.3679	0.4844	0.3292	0.9165	0.8744	0.8915	0.9589
	4	RoBERTa BCE	0.9813	**0.4749**	0.5359	0.3836	0.8891	0.8800	0.8901	0.9616
	5	RoBERTa FL	**0.9818**	0.4648	0.5524	0.4017	0.8839	**0.8807**	**0.9010**	0.9597
	6	RoBERTa pwBCE	0.9809	0.3541	0.4845	0.3284	0.9232	0.8741	0.8982	0.9575
	7	RoBERTa pwFL	0.9809	0.3612	0.4861	0.3297	0.9254	0.8734	0.8920	0.9597
	8	XLM RoBERTa	0.9790	0.3368	0.4680	0.3135	0.9230	0.8689	0.8859	0.9581
CCT	9	CCT	0.9505	0.3428	0.4874	0.3507	0.7983	0.8133	0.8307	0.9447
CNN	10	Freeze GloVe ResNet44	0.9526	0.4189	0.5591	0.4631	0.7053	0.8219	0.7876	0.9499
	11	Unfreeze GloVe ResNet44	0.9660	0.4566	**0.5958**	**0.4835**	0.7759	0.8421	0.8493	0.9540
	12	Unfreeze GloVe ResNet56	0.9639	0.3778	0.5098	0.3604	0.8707	0.8487	0.8445	0.9579
RNN	13	BiGRU	0.9748	0.3492	0.4762	0.3232	0.9036	0.8573	0.8616	0.9600
	14	BiLSTM	0.9754	0.3638	0.5089	0.3586	0.8761	0.8636	0.8758	0.9569
XLNet	15	XLNet	0.9800	0.3336	0.4586	0.3045	**0.9287**	0.8738	0.8834	0.9597

TAB. 4 – *Model performance results.*

4.1 Performances

According to Figure 1 and Table 4, in general on the AUROC metric, BERT, RNN and XLNet have better scores than the others. As the comments are pretty short on average (27

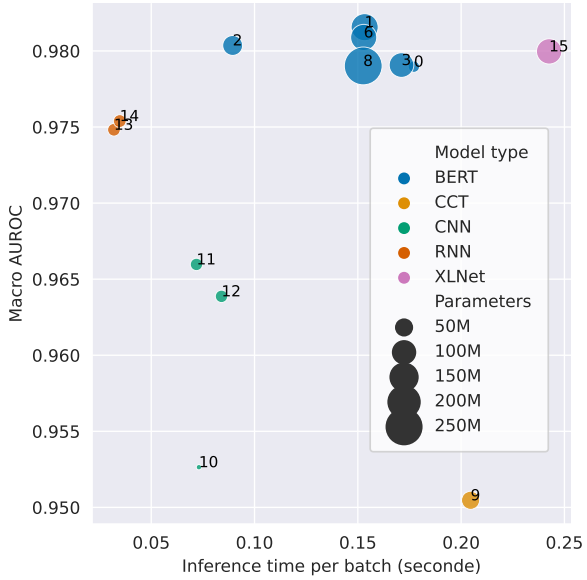


FIG. 1 – *Model performance, depending on the inference time per batch and the number of trainable parameters. The number associated with each point corresponds to the model id in Table 4. All models have a batch of 32 samples, except CCT, which uses a batch of 8.*

tokens), RNN keep in memory a large part of the message to make a good prediction. We probably would have seen a more significant performance gap between BERT and RNN if the comments had been longer. RoBERTa with Focal Loss models offers the best performance on biases and AUROC. All BERT, regardless of size and optimizations, have similar performance. A DistilBERT or an AlBERT is as good as a HateBERT. Even if we notice that RoBERTa with the Focal Loss gives less biased results on the identity groups than DistillBERT. XLM RoBERTa does not differ from the others: the model has learned about other languages, but it does not give an advantage in detecting hateful comments.

Across the models, Recall is often very high, and Precision remains low. In other words, the models are more sensitive to hateful comments but generate more false positives. On the other hand, these same models detect many more true positive hate comments.

Within BERT, if we look at the RoBERTa with the different training losses (BCE, pwBCE, FL, pwFL), we see relatively close scores at the end. None of the tested training losses improve the learning of the models compared to a simple BCE. We even note that the positive weights (pwBCE and pwFL) obtain worse F1 scores than the BCE or FL, but the Recall is 0.04 higher, and the accuracy is smaller by 0.1.

The Bi-GRU and Bi-LSTM have equivalent performance in terms of AUROC and F1 scores.

Nevertheless, we can show that all models have a little more difficulty classifying toxic comments and insults than explicit sexual comments.

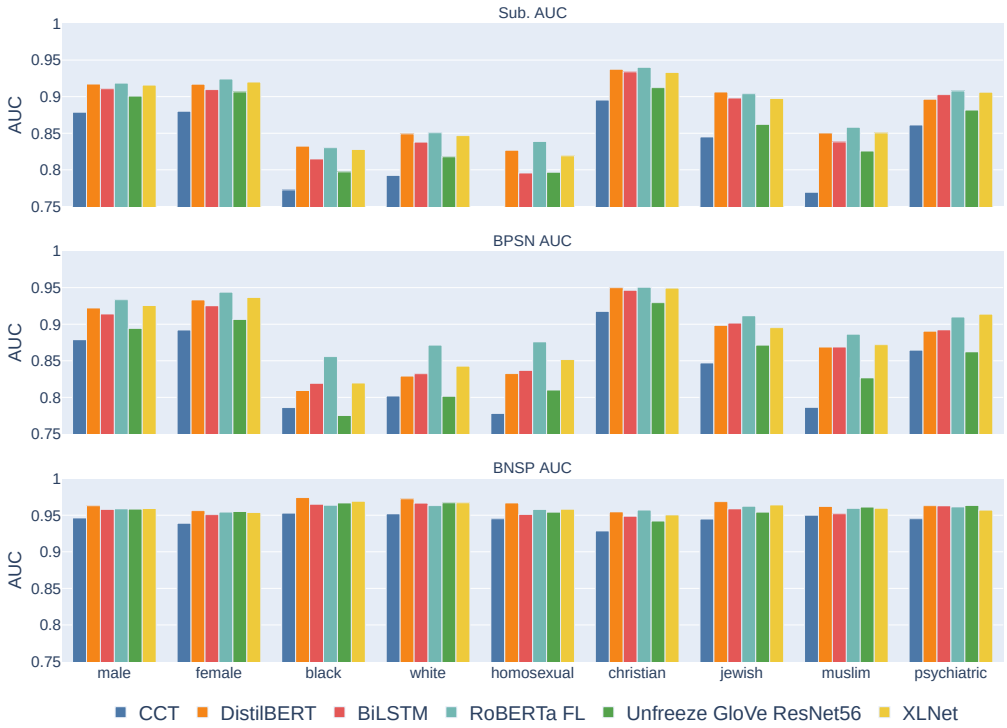


FIG. 2 – *Community-wise results for each bias-metrics on the toxicity class. Only the most relevant models are shown here for the sake of readability. Thus, we have kept only the BERT, RNN, and CNN models with the best performance on AUROC or on the AUC bias metrics.*

4.2 Bias

Overall, if we look at the results presented in Table 4, we see that the models have a GMB BNSP greater than 0.95. In other words, the models have no difficulty differentiating between hateful comments targeting a community and generic comments (without targeting a particular identity). On the contrary, we observe that the scores for GMB BPSN and GMB Sub are lower than those for GMB BNSP, often below 0.90. Thus, all the models present an association bias between identities and insults. They will tend to detect as being insults the positive comments about a community. But this bias depends on the model type.

From Table 4, we see that BERT and RNN models are generally less sensitive to this bias by having slightly higher GMB BPSN and GMB Sub. In contrast, convolution-based models such as CNN and CCT tend to be more sensitive to this association bias. CNN seek to capture patterns with convolutions.

From Figure 2, all models score worse on average on BPSN and Sub. AUC for the `black`, `homosexual`, `muslim`, and `white` communities compared to the other communities. For BPSN, this means that models have difficulty differentiating between insults that do not target identity and healthy comments about a community. Thus, these same models will tend to have

more association bias and detect healthy comments about these communities as toxic. For Sub. AUC, this means that when a comment targets an identity such as `black`, `gay`, `muslim`, or `white`, the models will have more difficulty distinguishing between hateful and non-hateful comments.

If we now look in more detail for each model and each identity, we notice again that *RoBERTa with FL*, *BiLSTM*, and *XLNet* are less affected by that bias than *Unfreeze GloVe ResNet56* and *CCT*. There is even a difference of 0.05 on the Sub AUC for comments targeting communities such as `jewish` or `muslim` between *RoBERTa with FL* and *Unfreeze GloVe ResNet56*. Similarly, there is a difference of up to 0.1 on the BPSN AUC for the `black`, `homosexual` and `muslim` communities. This shows that on these identities, which are particularly affected by hateful comments, the BERT, RNN, and XLNet models are less subject to association bias than the CNN and CCT.

4.3 Inference time

From Figure 1, with performances quite close to the BERT type models, RNNs have an inference time, per batch, 5 to 8 times smaller than BERT or XLNet. As expected, DistilBERT, with the smallest inference time tested in our study, is 2 times higher than Bi-GRU and Bi-LSTM, even if the performance difference is 0.005 in AUROC.

The CNN ends up with an inference time shorter than most BERT and larger than the longest RNN tested, but with much lower performance than RNNs or BERT. With the same inference time per batch, DistilBERT does better.

We also notice that freezing the embedding does not decrease the inference time, but decreased the model's performance.

Finally, the CCT offers disappointing performances with a very long inference time per batch, especially when we know that we have reduced the batch size from 32 to 8 for this particular model.

5 Conclusion

All BERTs have similar performance regardless of the size, optimizations, or language used to pre-train the models. More broadly, BERT, RNN, and XLNet have almost similar performance. RNNs are much faster at inference than any of the BERT tested. RNNs remain a good compromise between performance and inference time for multi-label detection of hateful comments. RoBERTa with Focal Loss models offers the best performance on biases and AUROC. However, DistilBERT combines both good classification performance and a low inference time per batch.

Even if the models are all affected by the bias of associating identities with toxicity, BERT, RNN, and XLNet are less sensitive to that than CNN and CCT.

References

Borkan, D., L. Dixon, J. Sorensen, N. Thain, and L. Vasserman (2019). Nuanced metrics for measuring unintended bias with real data for text classification. pp. 491–500.

Caselli, T., V. Basile, J. Mitrović, and M. Granitzer (2021). HateBERT: Retraining BERT for abusive language detection in English. In *Proceedings of the 5th Workshop on Online Abuse and Harms (WOAH 2021)*, Online, pp. 17–25. Association for Computational Linguistics.

Conneau, A., K. Khandelwal, N. Goyal, V. Chaudhary, G. Wenzek, F. Guzmán, E. Grave, M. Ott, L. Zettlemoyer, and V. Stoyanov (2020). Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, Online, pp. 8440–8451. Association for Computational Linguistics.

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

Dosovitskiy, A., L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby (2021). An image is worth 16x16 words: Transformers for image recognition at scale.

Hassani, A., S. Walton, N. Shah, A. Abuduweili, J. Li, and H. Shi (2021). Escaping the big data paradigm with compact transformers. *CoRR abs/2104.05704*.

He, K., X. Zhang, S. Ren, and J. Sun (2016). Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778.

Hinton, G., O. Vinyals, and J. Dean (2014). Distilling the knowledge in a neural network. In *Deep Learning and Representation Learning Workshop, NIPS*.

Isaksen, V. and B. Gambäck (2020). Using transfer-based language models to detect hateful and offensive language online. In *Proceedings of the Fourth Workshop on Online Abuse and Harms*, Online, pp. 16–27. Association for Computational Linguistics.

Lan, Z., M. Chen, S. Goodman, K. Gimpel, P. Sharma, and R. Soricut (2019). ALBERT: A lite BERT for self-supervised learning of language representations. *CoRR abs/1909.11942*.

Lin, T., P. Goyal, R. Girshick, K. He, and P. Dollar (2017). Focal loss for dense object detection. In *2017 IEEE International Conference on Computer Vision (ICCV)*, Los Alamitos, CA, USA, pp. 2999–3007. IEEE Computer Society.

Liu, P., W. Li, and L. Zou (2019). 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, Minnesota, USA, pp. 87–91. Association for Computational Linguistics.

Nguyen, D. Q., T. Vu, and A. Tuan Nguyen (2020). BERTweet: A pre-trained language model for English tweets. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, Online, pp. 9–14. Association for Computational Linguistics.

Pennington, J., R. Socher, and C. Manning (2014). GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Doha, Qatar, pp. 1532–1543. Association for Computational Linguistics.

Sanh, V., L. Debut, J. Chaumond, and T. Wolf (2019). Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. In *NeurIPS EMC2 Workshop*.

Sennrich, R., B. Haddow, and A. Birch (2016). Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, Volume 1, Berlin, Germany, pp. 1715–1725. Association for Computational Linguistics.

Swamy, S. D., A. Jamatia, and B. Gambäck (2019). Studying generalisability across abusive language detection datasets. In *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, Hong Kong, China, pp. 940–950. Association for Computational Linguistics.

Vaswani, A., N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin (2017). Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (Eds.), *Advances in Neural Information Processing Systems*, Volume 30. Curran Associates, Inc.

Yang, Z., Z. Dai, Y. Yang, J. Carbonell, R. Salakhutdinov, and Q. V. Le (2019). *XLNet: Generalized Autoregressive Pretraining for Language Understanding*. Red Hook, NY, USA: Curran Associates Inc.
English

Zhuang, L., L. Wayne, S. Ya, and Z. Jun (2021). A robustly optimized BERT pre-training approach with post-training. In *Proceedings of the 20th Chinese National Conference on Computational Linguistics*, Huhhot, China, pp. 1218–1227. Chinese Information Processing Society of China.

Résumé

La détection des commentaires toxiques sur les réseaux sociaux s'est avérée essentielle pour la modération de contenu. Dans cet article, nous comparons un large éventail de modèles sur un ensemble de données multilabels de discours haineux fortement biaisé. Nous prenons en compte dans notre comparaison le temps d'inférence, les performances et le biais à travers différentes métriques. Nous avons découvert que tous les BERT ont des performances similaires, indépendamment de leur taille, des optimisations ou du langage utilisé pour le pré-entraînement. BiLSTM reste un bon compromis entre la performance et le temps d'inférence. RoBERTa avec Focal Loss demeure le moins biaisé de tous. DistilBERT a le temps d'inférence le plus faible des BERT. Enfin, tous les modèles sont affectés par le biais d'association des identités à la toxicité. BERT, RNN et XLNet y sont moins sensibles que les CNN et les Compact Convolutional Transformers.

Evaluating The Effectiveness of Capsule Neural Network in Toxic Comment Classification using Pre-trained BERT Embeddings

Md Habibur Rahman Sifat[1], Noor Hossain Nuri Sabab[2] and Tashin Ahmed[3]

***Abstract*— Large language models (LLMs) have attracted considerable interest in the fields of natural language understanding (NLU) and natural language generation (NLG) since their introduction. In contrast, the legacy of Capsule Neural Networks (CapsNet) appears to have been largely forgotten amidst all of this excitement. This project's objective is to reignite interest in CapsNet by reopening the previously closed studies and conducting a new research into CapsNet's potential. We present a study where CapsNet is used to classify toxic text by leveraging pre-trained BERT embeddings (bert-base-uncased) on a large multilingual dataset. In this experiment, CapsNet was tasked with categorizing toxic text. By comparing the performance of CapsNet to that of other architectures, such as DistilBERT, Vanilla Neural Networks (VNN), and Convolutional Neural Networks (CNN), we were able to achieve an accuracy of 90.44%. This result highlights the benefits of CapsNet over text data and suggests new ways to enhance their performance so that it is comparable to DistilBERT and other reduced architectures.**

I. INTRODUCTION

In this age of cutting-edge technology, internet access is literally available to everyone. As it becomes more accessible, the cyber world becomes more susceptible to vulgar and abusive individuals. Therefore, it is becoming necessary for social media platforms to analyze and investigate inappropriate content on their platform. As more people join every day, it becomes increasingly difficult to manually police the Internet. Thus, an automated system that can detect toxic messages, quantify them, and then take the necessary actions is required.

This study aims to evaluate CapsNet against alternative neural network architectures for detecting potential toxicity in text data. We collected a dataset from Jigsaw (a Google think tank) via a 2020 Kaggle competition[a]. As performance evaluation of CapsNet is our primary concern, we have utilized googletrans library to translate non-English texts from the dataset and have not yet considered the issue from a multilingual perspective.

We implemented toxic text classification using CapsNet with pre-trained BERT embeddings (bert-base-uncased) and worked on a rich multilingual dataset. We worked on CapsNet to overcome CNN's subpar performance and achieved an accuracy of 90.44%, demonstrating CapsNet's performance over text data. Lastly, we implemented DistilBERT utilizing the same pretrained bert tokenizer as other methods (VNN and CNN) to achieve the highest accuracy of 93.24%. Our long-term objective is to make CapsNet's performance comparable to that of DistilBERT and other reduced architectures.

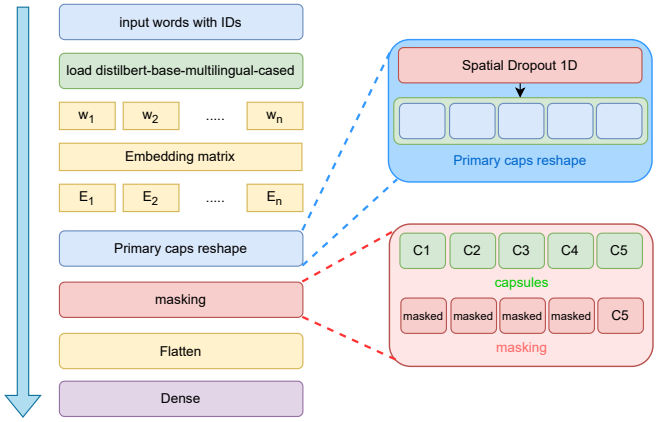


Fig. 1. Presented CapsNet architecture. The model takes word IDs as input and employs pre-trained BERT embeddings to extract context from text. A spatial dropout layer is applied to the BERT embeddings to prevent overfitting. The capsule layer receives the modified embeddings and learns to represent the input text as a collection of capsules, where each capsule represents a particular characteristic or attribute of the text. The capsule outputs are then fed into dense layers in order to learn higher-level text representations. The final dense layer generates the output prediction, which indicates the classification or label of the input text.

We implemented a number of neural network architectures, including VNN, CNN, CapsNet, and DistilBERT. We demonstrated that CapsNet extracts more features and performs better than CNN on text data.

II. RELATED WORKS

Classification of sentiment regarding toxicity has been massively studied in the recent years. Specially, researchers applied various machine learning algorithms on the social media data to identify the toxicity in online platform. Naguyen and Naguyen [8] proposed a DeepCNN and Bidirectional LSTM combined model which produce sentence wide feature embeddings from word level embeddings and achieved a good accuracy of 86.63%. Chu and June [2] achieved 93% accuracy on comment abuse classification problem using word embeddings by RNN with LSTM, CNN and character embeddings by CNN. Khieu and Narwal [5] showed CNN and LSTM based model which works for character level but did not works for sentence level.

Xie et al [10] proposed an ensemble method for the multilingual toxic comment classification where their model achieved an AUC of 0.9485 for the validation set. Some

[1]Md Habibur Rahman Sifat is with Department of Computing, The Hong Kong Polytechnic University, Hung Hom, Kowloon, Hong Kong. habib.sifat@connect.polyu.hk

[2]Noor Hossain Nuri Sabab is with Department of CSE, United International University, Dhaka, Bangladesh. nsabab@aol.com

[3]Tashin Ahmed is with the AI Team, Smart Studios, B'Kara, Malta. tashin@smartstudios.io

[a]https://www.kaggle.com/competitions/jigsaw-multilingual-toxic-comment-classification

TENCON 2023 - 2023 IEEE Region 10 Conference (TENCON) | 979-8-3503-0219-6/23/$31.00 ©2023 IEEE | DOI: 10.1109/TENCON58879.2023.10322429

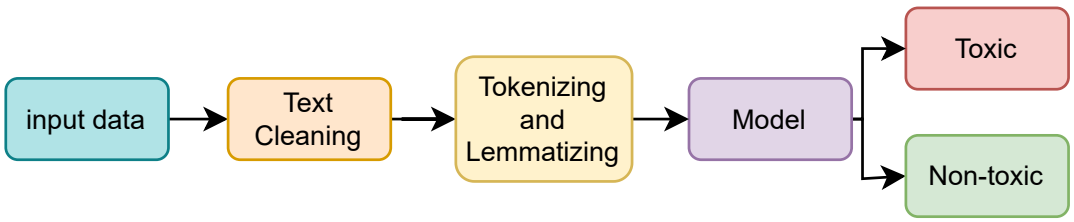


Fig. 2. A straightforward general structure of the experiment that have been performed on the text data.

researchers [1], [6] focused on regional language such as Brazilian Portuguese, Russian. Leite et al [6] achieved 75% of F1-score but using BERT model but it's struggling with data imbalancing and could not detect some comments. Xu et al [11] proposed an ensemble model that combines monolingual and multilingual models to achieve a better performance for both cases using XLM-RoBERTa multilingual fine-tuning model. Song et al [9] proposed a fusion model for the same dataset where they combined multiple loss function and multiple pre-trained model and their accuracy is 94.67.

Mazzia et al [7] designed an efficient capsule network which only require 2% of CapsNet parameters to get state-of-the-art result for three different datasets. They used the self-attention routing mechanism among the active capsules almost a fully connected network with additional branch of attention layer. They performed experiment on MNIST, MultiMNIST and smallNORB datasets and showed the effectiveness of CapsNet. Though all datasets are image based and showed the effectiveness of CapsNet in vision domain, we extend it for the text domain as well.

III. DATASET

The multiclass dataset for classifying toxic comments contained over 223,000 comments, with classes including toxic, severe toxic, obscene, threat, insult, and identity hate. The data was discovered to be extremely unbalanced, with the majority of comments being in English. In addition to analyzing sentiment and polarity with NLTK's Sentiment Intensity Analyzer (SIA), we measured readability with the Flesch reading ease, Automated Readability Index (ARI), and Dale-Chall readability measures. The Flesch reading ease was not found to be a reliable indicator of the toxicity of comments, whereas the ARI and Dale-Chall readability measures provided more insight into the dataset shown in Figure 3.

Negative sentiment is a score between 0 and 1 indicating negative or pessimistic emotions. The higher the score, the more negative the comment. The majority of comments in the dataset had low negativity, indicating that they were not toxic or negative. The most frequent negativity value was approximately 0.04, and few comments had a negativity greater than 0.80. When comparing negativity and toxicity, toxic comments had, on average, significantly more negative sentiment than non-toxic comments. At this point, the majority of non-toxic comments had a negativity value of 0, while toxic comments had the lowest negativity. This indicates that a comment with a negativity score of 0 is most likely harmless.

Positive sentiment is a score between 0 and 1 that represents positive or optimistic emotions. A higher score represents a more favorable comment. The majority of comments contained less positivity. Very few comments had a positivity score greater than 0.80. The most common positivity value was approximately 0.08. Comparing positivity and toxicity, we discovered that the graphs for toxic and non-toxic comments were comparable. This suggests that positivity is not a reliable indicator of comment toxicity.

Neutral sentiment refers to the text's lack of bias or opinion. It is a score between 0 and 1, with a higher score denoting a more neutral or impartial comment. The majority of comments were impartial and neutral. This suggests that comments were generally neutral and devoid of strong opinions. When contrasting neutrality and toxicity, non-toxic comments had, on average, higher neutrality values. At 1, there was an abrupt increase in the probability density of non-toxic comments, while the probability density of toxic comments was significantly lower at the same point. This suggests that comments with scores closer to 1 are more likely to be non-toxic.

Compound sentiment refers to a sentence's overall level of emotion. It is a score between -1 and 1, with higher scores indicating greater emotional intensity. The distribution of sentiment is uniform across the spectrum. It has a high standard deviation and random peaks throughout its range. Figure 4 shows detailed view of them. Some sample data are presented in Table I and a basic statistics of the dataset is shown in Table II.

IV. METHODOLOGY

In the initial stage of modeling, the data was cleaned to remove unnecessary elements such as punctuation, timestamps, and user information since it consisted of public comments. The data was then tokenized, which involved splitting the sentences into the smallest possible strings, with the aim of further text cleaning, such as lemmatization or converting the output to a data frame for improved use in a model. The study employed a pre-trained BERT-tokenizer for this purpose, which expedited the progress of the work.

For understanding how the processed data works with a neural network, we applied the BERT embeddings into a basic VNN, with 128 neurons in the Dense layer. We used Rectified Linear Unit (ReLU) as activation function and Binary Cross-entropy as the loss function, with Adam as the optimizer. The network provided us with an accuracy of 80.44%, which showed that the pre-trained BERT embeddings were really effective and also showed why it was considered a state-of-the-art language model.

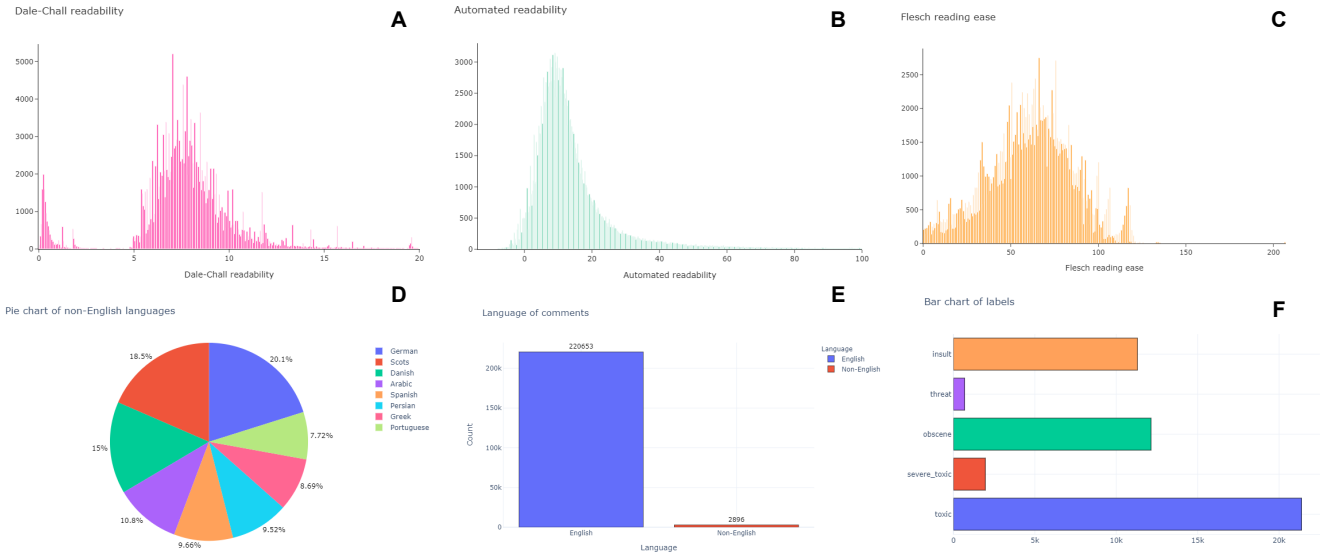


Fig. 3. Metrics to assess the readabilty or ease of understanding texts. A: Dale-Chall readability; B: Automated readability; C: Flesch reading ease. D: Non English language percentages; E: English and non-English language count; F: Toxic class counts.

id	comment_text	toxic	severe toxic	obscene	threat	insult	identity hate
0036621e4c7e10b5	Would you both shut up, you don't run wikipedia, especially a stupid kid.	1	0	0	0	1	0
0021fe88bc4da3e6	My Band Page's deletion. You thought I was gone. Deleting the comment I posted on your 'talk page'...	1	0	1	0	0	0
0007e25b2121310b	Bye! Don't look, come or think of comming back! Tosser.	1	0	0	0	0	0
00472b8e2d38d1ea	A pair of jew-hating weiner nazi schmucks.	1	0	1	0	1	1

TABLE I

SAMPLE DATA FROM THE KAGGLE COMPETITIONS DATASET, UTILISED IN THIS EXPERIMENT. IT INCLUDES A UNIQUE ID FOR EACH ENTRY, LEVEL AND CATEGORY OF TOXICITY, INCLUDING TOXIC, SEVERE TOXIC, OBSCENE, INSULT, AND IDENTITY HATE. 1S AND 0S INDICATE WHETHER THE COMMENT FALLS UNDER THE SPECIFIED CATEGORY.

	toxic	severe_toxic	obscene	threat	insult	identity_hate
0	202165	221587	211409	222860	212245	221432
1	21384	1962	12140	689	11304	2117
mean	0.1	0.01	0.05	0	0.05	0.01
std dev	0.29	0.09	0.23	0.06	0.22	0.1

TABLE II

COUNT OF TOTAL DATA FOR EACH CATEGORY. 1: REPRESENTS THE TOTAL NUMBER OF INSTANCES OF THE SPECIFIED CATEGORY. EACH DATA ENTRY (COMMENT) MAY BELONG TO MULTIPLE CATEGORIES (TABLE I).

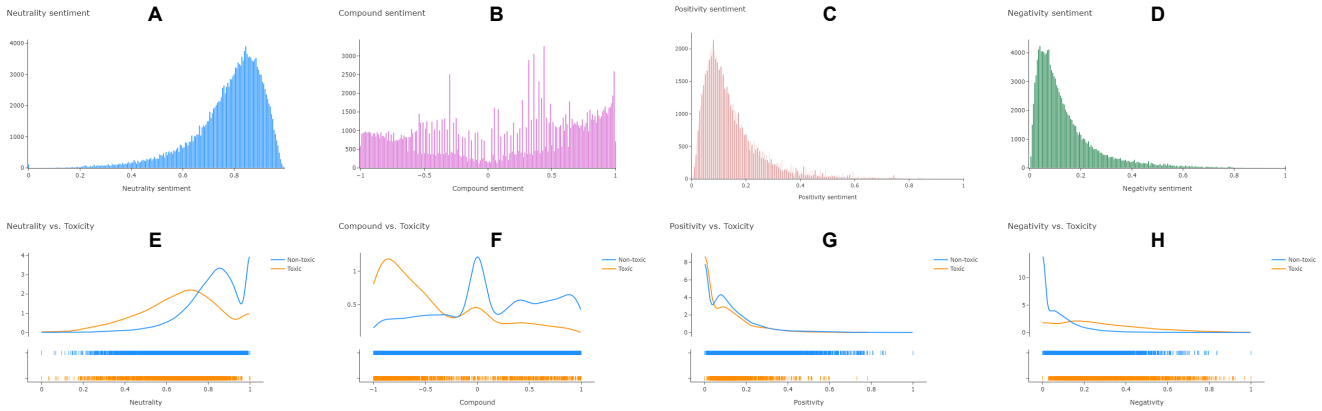


Fig. 4. **Sentiments scores.** A: Neutrality; B: Compound; C: Positivity; D: Negativity. **Comparative analysis against toxicity.** (E: Neutrality, F: Compound, G: Positivity, H: Negativity) vs Toxicity

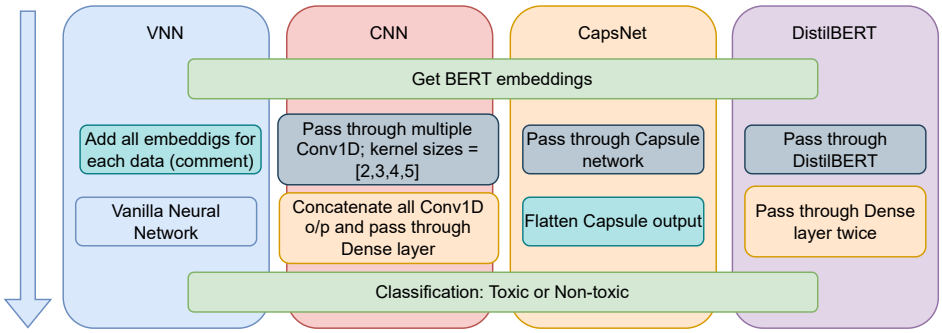


Fig. 5. Basic architecture of all the models (VNN, CNN, CapsNet, DistilBERT) that tested till now. Common layer for every architecture are the BERT embeddings and classifier.

Models	Accuracy	Loss	Val Accuracy	Val Loss
Vanilla Neural Network	85.98%	1.29%	80.44%	1.85%
Convolutional Neural Network	89.77%	0.95%	86.62%	1.03%
CapsNet	93.09%	0.80%	**90.44%**	1.45%
DistilBERT	97.54%	0.58%	**93.24%**	1.74%

TABLE III

MODELS PERFORMANCES IN TERMS OF 5 FOLD CROSS VALIDATION. 90.44% OF CV ACCURACY HAS BEEN ACHIEVED BY CAPSNET. AS A PRE-TRAINED MODEL DISTILBERT ACQUIRED 93.24%.

CNNs are a type of neural network generally used for image recognition problems, although the one-dimensional (1D) version of CNNs can also be used for text-related problems. CNN involves a process called convolution, which is a rather simple algorithm that involves a kernel (a 2D matrix) moving over the entire matrix (word embeddings), calculating dot products with each window along the way. In text classification, a 1D variant of convolution is used where the kernel moves in only one dimension. We used the pre-trained BERT embeddings as input, passed the embeddings through convolutional layers, and got the probability of the comment being toxic. We used four Conv1D layers with 64 neurons in each layer, passed those layers into four max-pool layers, then concatenated the max-pool layers to pass into a Dense layer of 64 neurons. As before, we used ReLU, binary cross-entropy, and Adam for the neural network architecture. We found out that even with a more complex model than the previous one, our accuracy improved very little, with an accuracy of 86.62% from CNN. For test cases, it was seen that it correctly classified all the four inputs that the VNN failed to do, proving that CNN understands spatial relationships better.

Since CNN couldn't provide a significant improvement in accuracy, we designed a CapsNet. CNNs can extract features, but they cannot understand the spatial and proportional relationships between objects in the image. CapsNet solves this problem by understanding the spatial relationships between words (in text) by encoding additional information. A lot of the intuition behind how and why CapsNets work is linked to image recognition, but it can also be used for text-based problems. We used a capsule layer with 5 capsule numbers and 5 dimensions for the model. The layer was then passed through Flatten to change the shape of the data dimension to correctly pass through the Dense layer, with 128 neurons, and same organization as the other models.

Our CapsNet architecture has been built with a BERT-based embedding layer as the underlying structure for text classification tasks. The model accepts word IDs as input and employs pre-trained BERT embeddings to extract context from text. A spatial dropout layer is applied to the BERT embeddings to prevent overfitting. The capsule layer receives the modified embeddings and learns to represent the input text as a collection of capsules, where each capsule represents a particular characteristic or attribute of the text. The capsule layer refines capsule activations through multiple routing iterations by giving greater weight to capsules with higher agreement. The capsule outputs are then fed into dense layers in order to learn higher-level text representations. The final dense layer generates the output prediction, which indicates the classification or label of the input text. For training, the model is compiled with an appropriate optimizer and loss function. This architecture permits more expressive and hierarchical representations of the input text, thereby improving the model's ability to accurately classify text. Figure 1 shows the high-level diagram of the implemented CapsNet.

V. RESULTS

The accuracy acquired is 90.44%, which was higher than the CNN model. But it also couldn't predict two of the four test cases correctly which proved that CapsNet is not well-suited for text-related data. In CNN, neurons are activated to detect specific features, where they do not consider the properties of features like size, orientation, color, velocity, and so on. It is not trained on relationships between features and because MaxPooling in CNN, it loses information. MaxPooling is performed to achieve translation variance. A capsule in a CapsNet is made up of a layer of neurons that perform internal calculations to forecast the existence and the instantiation parameters of a specified feature at a particular spot.

DistilBERT is a lighter version of BERT [4] which uses

fewer weights and achieves similar accuracy on several tasks with much lower training times (40% fewer parameters than bert-base-uncased, runs 60% faster). BERT (Bidirectional Encoder Representations from Transformers) is a paper published by researchers at Google, which caused a great stir in the NLP community as it became the state-of-the-art on several NLP tasks. BERT's key technical innovation was applying the bidirectional training of Transformer, a popular Attention model, to language modeling. This was in contrast to previous efforts which looked at a text sequence either from left to right or combined left-to-right and right-to-left training (such as LSTMs).

The paper's results showed that a language model which was bidirectionally trained can have a deeper sense of language context and flow than single-direction language models. In the paper, the researchers detailed a novel technique named Masked Language Model (MLM) which allows bidirectional training in models in which it was previously impossible. We used the BERT embeddings again, used them in our Transformer layer, which was then passed to a Dense layer with 500 neurons. We used a Dropout of 0.1 to avoid overfitting of data. We kept all the parameters the same as the other models previously described and achieved an accuracy of 93.24% (Table III), although it misclassified one of the test cases, which might be because we trained the model for 50 epochs, like the rest of the models. Figure 5 shows the workflow of all the models applied during this work.

VI. CONCLUSION

We investigated the dataset and characterized its sentiment, polarity, and readability. In addition, we have implemented a number of neural network models and demonstrated their performance using BERT embeddings for each of their backbones. Finally, a capsule neural network was designed for multiclass text classification, obtaining promising results that outperformed a CNN architecture with similar parameter counts. In addition, the DistilBERT model achieved the best results for this task, which inspired us to set a target and create a simple and effective model to address the current problem.

VII. FUTURE WORKS

The work is still in progress, and we choose DistilBERT as the benchmark in order to improve CapsNet's text data performance over CNN. Instead of using ReLU, which is quite similar to ELU [3] but tends to converge cost to zero much more slowly, we will keep investigating using other hyperparameter adjustment techniques, which will likely yield more accurate finding during CapsNet training. In order to conduct a more thorough comparison and analysis and identify CapsNet's shortcomings, our research also involves adding Attention layers to the simulation and doing research with other models, such as LSTM and so on. As the dataset is multilingual and it is essential to solve the task from a multilingual perspective, we will work on a non-English or multilingual pre-trained model to improve the CapsNet architecture without relying on translation packages.

REFERENCES

[1] Darya Bogoradnikova, Olesia Makhnytkina, Anton Matveev, Anastasia Zakharova, and Artem Akulov. Multilingual sentiment analysis and toxicity detection for text messages in russian. In *2021 29th Conference of Open Innovations Association (FRUCT)*, pages 55–64. IEEE, 2021.

[2] Theodora Chu, Kylie Jue, and Max Wang. Comment abuse classification with deep learning. *Von https://web. stanford. edu/class/cs224n/reports/2762092. pdf abgerufen*, 2016.

[3] Djork-Arné Clevert, Thomas Unterthiner, and Sepp Hochreiter. Fast and accurate deep network learning by exponential linear units (elus). *arXiv preprint arXiv:1511.07289*, 2015.

[4] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[5] Kevin Khieu and Neha Narwal. Detecting and classifying toxic comments. *Web: https://web. stanford. edu/class/archive/cs/cs224n/cs224n*, 1184, 2017.

[6] Joao A Leite, Diego F Silva, Kalina Bontcheva, and Carolina Scarton. Toxic language detection in social media for brazilian portuguese: New dataset and multilingual analysis. *arXiv preprint arXiv:2010.04543*, 2020.

[7] Vittorio Mazzia, Francesco Salvetti, and Marcello Chiaberge. Efficient-capsnet: Capsule network with self-attention routing. *Scientific reports*, 11(1):14634, 2021.

[8] Huy Nguyen and Minh-Le Nguyen. A deep neural architecture for sentence-level sentiment classification in twitter social networking. In *Computational Linguistics: 15th International Conference of the Pacific Association for Computational Linguistics, PACLING 2017, Yangon, Myanmar, August 16–18, 2017, Revised Selected Papers 15*, pages 15–27. Springer, 2018.

[9] Guizhe Song, Degen Huang, and Zhifeng Xiao. A study of multilingual toxic text detection approaches under imbalanced sample distribution. *Information*, 12(5):205, 2021.

[10] Gaofei Xie. An ensemble multilingual model for toxic comment classification. In *International Conference on Algorithms, Microchips and Network Applications*, volume 12176, pages 429–433. SPIE, 2022.

[11] Jian Xu and Yuqing Zhai. A toxic comment classification model based on ensemble. In *Journal of Physics: Conference Series*, volume 1873, page 012080. IOP Publishing, 2021.

Handling Bias in Toxic Speech Detection: A Survey

TANMAY GARG, SARAH MASUD, THARUN SURESH, and TANMOY CHAKRABORTY, IIT Delhi, India

Detecting online toxicity has always been a challenge due to its inherent subjectivity. Factors such as the context, geography, socio-political climate, and background of the producers and consumers of the posts play a crucial role in determining if the content can be flagged as toxic. Adoption of automated toxicity detection models in production can thus lead to a sidelining of the various groups they aim to help in the first place. It has piqued researchers' interest in examining unintended biases and their mitigation. Due to the nascent and multi-faceted nature of the work, complete literature is chaotic in its terminologies, techniques, and findings. In this article, we put together a systematic study of the limitations and challenges of existing methods for mitigating bias in toxicity detection.

We look closely at proposed methods for evaluating and mitigating bias in toxic speech detection. To examine the limitations of existing methods, we also conduct a case study to introduce the concept of *bias shift* due to knowledge-based bias mitigation. The survey concludes with an overview of the critical challenges, research gaps, and future directions. While reducing toxicity on online platforms continues to be an active area of research, a systematic study of various biases and their mitigation strategies will help the research community produce robust and fair models.[1]

CCS Concepts: • **General and reference** → **Surveys and overviews**; • **Information systems** → *Social networks*; • **Social and professional topics** → **User characteristics;**

Additional Key Words and Phrases: Toxic speech, hate speech, social networks, unintended bias, bias mitigation, bias shift

ACM Reference format:
Tanmay Garg, Sarah Masud, Tharun Suresh, and Tanmoy Chakraborty. 2023. Handling Bias in Toxic Speech Detection: A Survey. *ACM Comput. Surv.* 55, 13s, Article 264 (July 2023), 32 pages.
https://doi.org/10.1145/3580494

[1]**Disclaimer:** This article includes examples of toxic speech that contain some profane words. These examples are only included for contextual understanding. We tried our best to censor vulgar, offensive, or hateful words. We assert that we do not support these views in any way.

T. Garg and S. Masud contributed equally.

We would like to thank the support of Prime Minister Doctoral Fellowship (SERB India), Ramanujan Fellowship (SERB, India), and the Wipro Research Grant.

Authors' address: T. Garg, S. Masud, T. Suresh, and T. Chakraborty, IIIT Delhi, India; emails: {tanmay17061, sarahm, tharun20119, tanchak}@iiitd.ac.in.

0360-0300/2023/07-ART264 $15.00
https://doi.org/10.1145/3580494

1 INTRODUCTION

Online social networks (OSNs) have enabled a thriving ecosystem where people from diverse backgrounds can share opinions and ideas. Such online forums are safe spaces for many marginalized groups to support each other and gain a sense of community in the face of discrimination. However, anti-social users often use OSNs to harass and intimidate others. Such behavior manifests in content aimed at harming individuals or groups based on personal attributes such as race, gender, and ethnicity. At its extreme, hate speech (a type of toxic speech) can lead to incidents of offline violence [28, 59]. It should be noted that such incidents of violence and hate crime do not occur in isolation. Often negative stereotyping and hostility in the real world get heightened during online interactions. Meanwhile, online hate speech and doxing [102] promote offline brutality and even genocides. The online and offline propagation of toxic behavior is a vicious cycle. Such behavior often hampers the ability of marginalized groups to share their opinions freely and further isolates them [83, 106]. Therefore, the real-world impact of harmful online content and its multifaceted nature have galvanized academic and industrial research toward the early detection and mitigation of such content.

Toxic speech. Throughout this survey, the term toxic speech will be used as an umbrella term to refer to any form of content, including but not limited to hate speech, cyberbully, abusive speech, misogyny, sexism, offense, and obscenity. We follow the definition of toxic speech as given by Dixon et al. [24]: "rude, disrespectful or unreasonable language that is likely to make someone leave a discussion" as an umbrella definition. The literature surveyed here is distributed among various forms of toxic speech, known to adopt different and sometimes misleading terms to refer to equivalent classes [33]. The subjectivity in defining toxicity can also be attributed to the lack of an in-depth understanding of toxicity through the lens of social and psychological science. Machine learning researchers and practitioners often overlook the root cause of toxic behavior on the Internet [95]. Such root cause analysis can help understand the power dynamics of hate [66], the prevalent discrimination, and its evolution on the Internet [30, 58]. Recent studies in implicit hate speech [27, 42] attempt to bridge this gap; however, such practice in computational methods for toxicity detection is not yet well adopted.

Vulnerable groups. In relation to toxic speech, there is always a group (or an individual from the group) at the receiving end of all the negativity and hate. Such groups can be termed as vulnerable/target/protected groups. For bias evaluation, they may be even referred to as subgroups or identity groups. Each of these groups can be identified with negative phrases that are used to refer to the individuals of the particular group. Such phrases are called "identity terms." For example, when analyzing hate speech against Jewish people, "k*k*" is a very commonly used slur.

Unique characteristics of online toxic speech. Toxicity detection is a highly subjective task, and context plays a crucial role in determining whether the content can be flagged as toxic. Different online platforms and legal agencies use varying definitions of toxicity. Meanwhile, existing datasets display a variety of biases via their collection pipeline [24, 113] and are subject to unreliable annotations [81]. This article explores why these characteristics render toxic speech vulnerable to various unintended biases and the methods proposed to handle them.

Scope of the survey. Bias mitigation methods in **natural language processing (NLP)** have been extensively studied [7, 36, 110]. Consequently, the mitigation techniques in NLP have been helpful in tasks such as textual entailment [100] or reading comprehension [35]. It also motivated researchers to apply bias mitigation for toxic speech detection. While the respective methods have progressed, the results are not as effective as expected. The argument goes back to the subjective/objective nature of tasks like toxicity detection vs. textual entailment. It inspired us to conduct a survey that can help better understand the current state-of-the-art bias mitigation in

Table 1. List of Popular Toxic Speech Datasets in the Study of Bias as Discussed in This Article

Dataset	Sampling	Lexical		Annotation	Racial		Gender	
	E	E	M	E	E	M	E	M
Founta [34]	[69, 113]	[113]	[120]		[19, 20, 44]	[85, 116, 120]	[44]	[70]
W&H [109]	[4, 69, 113]	[113]		[64, 108]	[20, 64]			[70, 113]
Wulczyn [115]	[78, 113]	[113]	[5, 24]	[2, 111]				
Davidson [21]			[5]	[64]	[20, 44, 64, 85]	[116]	[44]	
Waseem [108]				[64, 108]	[20, 64]			[70]
GHC [48]			[49]					
Stormfront [22]			[49]					
Evalita2018 [29]								[68]

"E" (Evaluates) indicates when a dataset was used to establish the presence of bias. "M" (Mitigates and Evaluates) indicates if debiasing techniques were performed on the dataset. Here, sampling, lexicon, and annotation are sources of bias, whereas race and gender are targets of bias. Note that this table is a representative sample of the literature surveyed here and is not a comprehensive set.

toxicity detection. The scope of our study is not to discuss the comprehensive research of biases in NLP; instead, we present a thorough analysis of the methods that study *bias as applied to the case of automatic toxicity detection*. Additionally, several surveys have already examined the existing literature in modeling the toxicity detection task [32, 89]. We do not survey all the existing toxicity detection methods; instead, we focus on a subset of them, exploring and mitigating bias in toxic speech detection. Meanwhile, Yin and Zubiaga [117] surveyed the literature addressing the robustness of hate speech detection methods and addressed the subject of bias in hate speech detection. While their discussion remained general commentary, we aim to develop an extensive understanding of these methods.

To begin with, we develop a taxonomy of bias based on the sources and targets of harm. Each bias mitigation method can be applied to one or more data transformation stages of a **machine learning (ML)** pipeline [98]. Subsequently, Table 1 provides a summary of popular hate speech datasets and their usage for the study of evaluating and mitigating bias in toxic speech detection. As a part of our literature survey, we also review the reproducibility of existing debiasing methods employed for toxicity detection. The details of the experiments are provided in Appendix C. Meanwhile, we also discover a compelling phenomenon of bias shift in knowledge generalization-based methods and provide a short description of the same in Appendix B.

Survey methodology. Following Yin and Zubiaga [117], we considered Google Scholar as the primary search engine to curate relevant papers. We focused on toxicity debiasing studies done within the past five years and mainly looked at research published in 2016 onwards. We started with relevant keywords such as "bias," "toxic speech," "abusive speech," and "hate speech," shortlisting a seed set of papers through their abstracts. Papers were also collected from recent proceedings of relevant data mining, NLP and web-science conferences (ACL, EMNLP, NAACL, AAAI, WebSci, ICWSM, etc.), journals (TACL, TKDD, PLOS, etc.), and workshops (WOAH etc.). We also visited the citing and cited papers of the seed papers to locate relevant papers further. This shortlisting process was done between September–November 2021.

2 CATEGORIES OF BIAS

All machine learning models, including toxicity detection, assume some bias in the data to perform predictions [24]. However, we do not intend the toxicity prediction to vary based on the speaker's racial background, for example. If a model exhibits such bias, then we call it *unintended bias*. In the rest of the article, we use "bias" to refer to unintended bias in toxic speech detection.

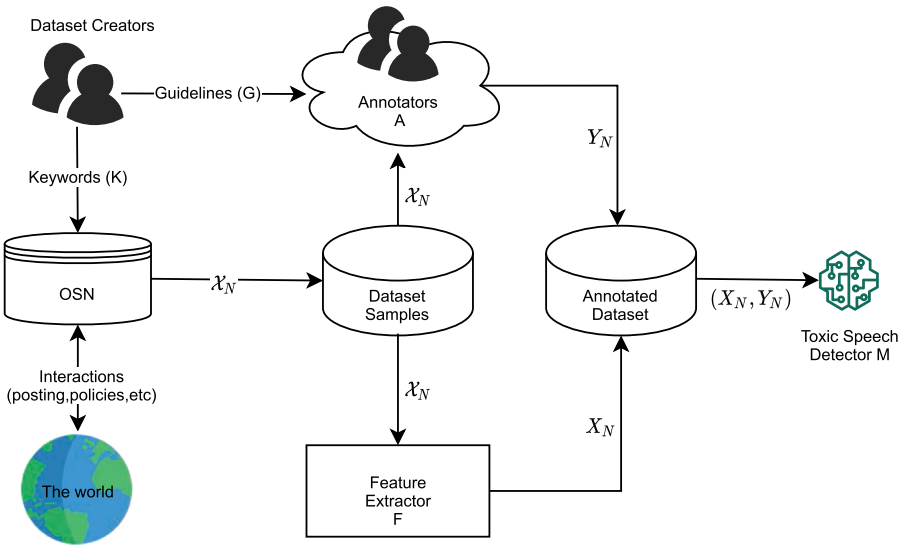


Fig. 1. The pipeline of a toxic speech detection model can be visualized as a sequence of data transformations: (i) the OSN sampler API ($S : K \rightarrow \mathcal{X}_N$) takes as input a set of keywords K (possibly empty) and returns a set of samples $\mathcal{X}_N$; (ii) the annotation function ($A : \mathcal{X}_N, G \rightarrow Y_N$) converts $\mathcal{X}_N$ and the annotation guidelines G to a sequence of annotations Y_N; (iii) the feature extraction function ($F : \mathcal{X}_N \rightarrow X_N$) converts $\mathcal{X}_N$ to a sequence of feature vectors X_N. (iv) Finally, the function ($M : \mathcal{X}_N, Y_N \rightarrow Y'_N$) aims to predict the toxicity label. Biases based on the source of harm are associated with these transformations as (i) sampling bias with S, (ii) lexical bias with (F, A), and (iii) annotation bias with A. Meanwhile, the biases based on the target of harm manifest when the prediction Y'_N of the model M are analyzed.

Background on NLP Bias. In cognitive science, bias is assumed to be a shortcut (valid/invalid)—our brain resorts to when informing our actions and interactions with others. Biases can develop due to a limited worldview and repeated exposure to similar behavior in the surrounding [43]. Stereotypes get further amplified due to the formation of filter bubbles [6] and echo chambers. When such biased interactions are employed to train natural language models, they learn stereotypical statistical associations prevalent on the Internet [15]. The seminal work by Bolukbasi et al. [10] led the initiative of establishing gender (male vs. female) bias in non-contextual embeddings trained on large-scale web corpora. This work prompted similar studies [7, 15] of bias in NLP towards other social/personal attributes.

Blodgett et al. [7] broadly classified the NLP biases as either allocational or representational harm. Sun et al. [96] further granulated representation harm into denigration, stereotyping, recognition, and under-representation when evaluating various NLP tasks and their associated language models for mitigating gender bias. In a tangential approach, Shah et al. [91] and Kumar et al. [52] looked at biases in NLP through the lens of data and modeling pipelines. Shah et al. [91] proposed a pipeline to highlight the origin of various biases and their eventual outcome disparity. Kumar et al. [52] largely classified the biases as related to either data source, label/annotation, and embedding representation.

Relation to NLP Bias. A simple bias mapping pipeline for training a generic NLP model is depicted in Figure 2. Comparing the training pipelines in Figures 1 and 2, we can observe how the first step (data collection and sample) in both workflows is primarily the source of sampling bias. In the case of toxicity detection, biased sampling is performed to gain a higher percentage of toxic samples. Consequently, in broader NLP modeling, it occurs due to a skewed ratio in

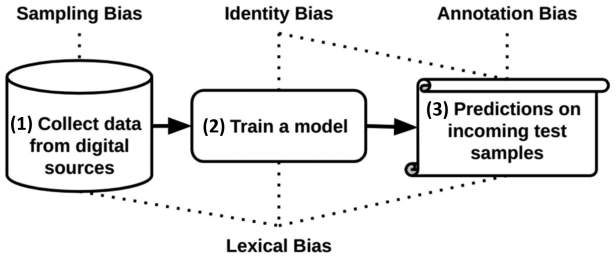


Fig. 2. The pipeline of training and evaluation of NLP models can be visualized as a sequence of (1) data collection, (2) training of models, and (3) employing the models on incoming test samples. Here, collecting data from digital sources can introduce sampling and lexical biases (if specific web pages/digital footprints are ignored). Model training can also introduce linguistic bias when frequently picking co-occurring phrases. Training processes and skewed datasets can introduce identity-based prejudice (race, gender, age, political affiliation, etc.). If predictions on downstream NLP tasks happen on human-labeled gold datasets, then it can be a source of annotation and sampling biases (not depicted here). Both linguistic and identity-based biases manifest during the model evaluation and in-the-wild testing.

terms of quality and quantity of digital footprints of specific topics (such as content in support of LGTBQ vs. against them). Sometimes, the training of NLP models can be unsupervised; in such cases, the annotation bias in the pipeline shifts to the last stage when it is employed on downstream tasks that use gold-label datasets to establish performance. All the shortcomings of human annotation apply to various NLP tasks. Some are more static and objective, leading to less bias and disagreements. Meanwhile, others can be highly subjective and ephemeral. As a result of various sampling and annotation biases, we observe the prevalence of identity-based harm introduced due to spurious lexical correlations. We term the bias/prejudice against a person's demographic or psychographic characteristic or any identifiable physical or mental characteristic as *identity bias*. As such, prejudice against specific identity groups can be analyzed separately based on the target of harm. Inspired by various approaches, we develop the following taxonomy based on the (i) sources and (ii) targets of harm. Biases studied in this survey are not unique to the task of toxicity detection; instead, they are an extension of the biases in NLP applications.

Based on *sources* of harm: We take inspiration from Suresh and Guttag [98] to group the surveyed methods into categories based on the source of downstream harms during the data collection process. The authors defined the process, consisting of: selecting a *population*, selecting and measuring *features*, and *labels* to use. We study categories of bias according to the transformations related to these steps (described in Figure 1) as: sampling, lexical, and annotation bias.

Based on *targets* of harm: The next three categories of bias in toxic speech are each dedicated to a target group of downstream harm (Figure 3): (i) racial bias, (ii) gender bias, and (iii) psychographic bias like political affiliations [45]. However, the study of biases based on psychographic attributes (grouping individuals w.r.t. their beliefs and interests) is yet to gain popularity. Through this survey, we hope to encourage future exploration of bias categories tied with psychographic attributes.

Other categorizations of bias. Some of the papers surveyed here present implicit vs. explicit bias categorization. We refer the readers to Reference [56] for an understanding of this bias stratification. Blodgett et al. [7] use a taxonomy that categorizes between *allocational* and *representational* harms. While this taxonomy is useful to segment existing papers based on their motivation of handling bias, the taxonomy we developed lets us approach the proposed methodologies from an application point of view.

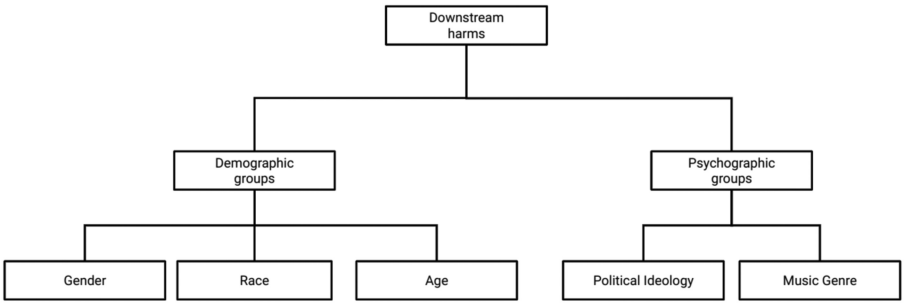


Fig. 3. A taxonomy of bias based on the downstream harm. The harms can be inflicted through the real-world deployment of a toxic speech detection system.

3 BIASES BASED ON SOURCE OF HARM

Overview. This section will cover three main categories of biases that stem from the data processing step. We begin our discussion with sampling and annotation bias and then cover lexical bias as the consequence of sampling and annotation. Sampling bias occurs at the first step of data collection. The workaround employed to obtain a more significant percentage of toxic comments against vulnerable communities eventually manifests as bias against the very community it intends to protect.

Such a dataset may require fine-grained labels obtained at large scale by human-aided annotations. However, performing annotations for subjective tasks such as labeling toxicity is tricky, because there is no standard definition of what can be considered toxic. Here, the cognizance of the annotators about the existing discrimination and current socio-political status quo is essential in obtaining valuable annotations. Especially when looking into toxicity against a particularly vulnerable group such as sexual orientation, it is necessary to consider the annotators' political, religious, and social views before annotations. It should be noted that such social beliefs need not be independently formed. Religion, for example, is not a personal belief system but can be culturally reinforced, sometimes acting as a proxy for race and ethnicity.

Building further on the context of sexual orientation, if the annotators are provided with metadata and background of the speakers participating in the comments, then it can be a double-edged sword. On the one hand, it can provide the context to correctly label such comments as non-toxic despite containing seemingly offensive terms. On the other hand, it can trigger latent biases in the annotators. They can willfully provide erroneous labels and label the interaction as hateful and harmful, leading to suppression of the voices of the minority.

Once the data is collected and annotated, it is ready for pre-processing and modeling. Once modeled, we can retrospectively analyze the models for downstream harms and biases it incorporates against different identity groups such as race, gender, or both. One such bias is lexical. Lexical bias can be evaluated at both dataset (source of harm) and modeling level (downstream harm). For this survey, we include it with the source of harm, as lexical biases may or may not be identity-specific and can be generally spurred from the presence/absence of words in both toxic and non-toxic labels. It should be noted that all biases we discuss in the coming sections, i.e., sampling, annotation, and lexical biases, directly/indirectly reflect the varying linguistic and social attributes and subjective nature of toxic content. An abbreviation of various terms used in our discussion of biases is enlisted in Table 2. A complete description of the various bias metrics and their usage is provided in Appendix A.

Table 2. Abbreviations and Expansions Used throughout the Article

Abbreviation	Expansion
SBET	Synthetic Bias Evaluation set
BSW	Bias Sensitive Words
SOC	Sampling and Occlusion
L_{CE}	Cross-entropy Loss
AAE	African-American dialectal English
NHB (NHW)	Non-Hispanic Black (White)
FPR (FNR)	False Positive (Negative) Rate
FPED (FNED)	False Positive (Negative) Equality Difference
pAUCED	pinned AUC Equality Difference
subAUC	Subgroup AUC
BPSN (BNSP)	Background Positive (Negative) Subgroup Negative (Positive) AUC
GMBAUC	Generalized Mean of the Bias AUCs
pB	pinned Bias
FPR (TPR)	False (True) Positive Rates
NPV (PPV)	Negative (Positive) Predictive Value
WEAT	Word Embedding Association Test
SEAT	Sentence Encoding Association Test

3.1 Sampling Bias

The first step towards toxicity detection is to curate data samples from across the web. Among social media content, harmful content is not a majority class. On Twitter, they cover no more than 3% of the tweets [34]. A random sampling of data thus requires going through a considerable amount of non-toxic content to get a substantial amount of toxic samples. Hence, datasets are curated by adopting heuristics that achieve a higher density of toxicity in the samples. These heuristics can introduce spurious correlations of linguistic features with toxicity labels if not tracked, therefore termed as *sampling bias*. In this section, we will give an overview of the two broad methods of sampling data from web and how to measure and mitigate the sampling bias.

Sampling techniques and effect of text source for toxicity datasets. In *boosted random sampling*, heuristics are applied to increase the density of harmful content in an initial random sample. This sample is usually labeled for toxicity, and the keywords or user names in these samples are used to extract more samples. Meanwhile, *topic-biased sampling* [84] starts by acquiring samples based on a predetermined set of keywords (hate lexicons) and socio-political topics that are commonly known to be toxic. Dataset creators have popularly used topic-biased sampling to quickly acquire samples with a higher toxicity concentration. Wiegand et al. [113] studied six datasets (three boosted and three topic-wise sampled) and used an abuse lexicon to mark examples as explicitly or implicitly abusive. They found the boosted randomly sampled datasets to contain a lesser concentration of overall toxicity; out of them, a higher concentration was explicit abuse. Moreover, a higher explicit abuse ratio was seen to aid in downstream performance, as *verbal cues for explicitly toxic labels are easier to model.* Their work was further investigated by Razo and Kübler [78] by conducting sampling experiments on the same underlying dataset. While broadly consistent with previous studies, they discovered that, all things being equal, the text source has a more significant impact on the quality of the toxic dataset curated than the underlying sampling strategy.

Metrics to measure sampling bias. To capture the topical and lexical biases in the curated datasets, Ousidhoum et al. [69] developed two bias metrics. These measures make use of predefined hate lexicons $w' \in W'$. While B_1 captures the average similarity between all topical words and

the hate lexicons, B_2 captures how likely each topic contains a keyword. The robustness in B_1 scores on increasing topic count indicates the new topic words to be similar to the keywords, pointing towards a topic bias. For instance, the Waseem & Hovy dataset (referred to as the W&H dataset henceforth) [109] showed robustness to topics, which can be attributed to its high topic and user biases [4, 113]. An Arabic dataset [3], which was curated based on religious keywords, gave comparable B_1 values for all three target attributes, suggesting its coverage of origin and gender-based topics as well. The authors further showed that these biases and variations in sampling techniques impact cross dataset generalization. In short, one can say that models trained on a dataset sampled for a set of topics do not generalize.

Mitigation via proxy topic-sampling. Van Rosendaal et al. [105] created a dataset by automating the pre-selection of keywords to reduce sampling bias. They used Reddit to find keywords from posts with more or less equal amounts of upvotes and downvotes. Such controversial topics can be used as a proxy to collect toxic speech from OSNs [40]. A small sample from the tweets filtered based on these keywords was verified as more toxic than an equal-sized sample of unfiltered tweets. However, this difference was minimal, suggesting the inability of the proposed method to create a dataset with the desired density of toxic comments. Moreover, Van Rosendaal et al. [105] did not validate their initial claim of a lower bias in the filtered dataset.

Mitigation via data mixing. Topic/author bias degrades downstream performance—the W&H dataset [109] is a *case in point.* Wiegand et al. [113] noticed that removing top PMI terms from the W&H dataset for the toxic labels led to a considerable drop in classification performance, suggesting a high topic bias. The dataset also contained high author bias, with more than 70% sexist tweets from only two authors and about 99% racist tweets from a single author. This high author bias can be attributed to the improvements in performance when encoding the user meta-information as features [63, 75]. It begs whether the classification model trained on the W&H dataset learns to discriminate the hate styles or the hateful authors' styles. Arango et al. [4] tried to correct the author bias in this dataset by limiting the number of toxic comments to 250 per user while augmenting more toxic comments from the Davidson dataset [21]. They found a substantial increase in generalization on the unseen Hateval dataset [46]. *However, we note that they did not account for the topical homogeneity between the training and test datasets that could have led to an increase in performance.*

3.2 Annotation Bias

As toxicity is subjective, it is challenging to train machine learning models to learn what can be potentially toxic. To obtain these labels, we require large-scale human-aided annotations. Fortuna et al. [33] showed that toxic speech datasets use ambiguous and misleading labels to refer to similar underlying categories introducing subjectivity into the annotation process. The subjectivity and discrepancy in the labeling process can lead to *annotation bias.* In this section, we will broadly discuss how the quality of data labels is impacted by the annotation guidelines and the annotator's prejudice. It should be noted that there is no direct way of mitigating annotating bias via modeling alone. Obtaining unbiased annotations requires sensitization of the annotators and the society at large. We hope this section highlights the ambiguous and challenging nature of generating standard labels for toxicity.

The effect of guidelines on the quality of annotations. Annotation pipelines need to consider the subjective nature of hate speech [82]. Ross et al. [81] explored the effect on the reliability of annotations before and after providing the guidelines. Both annotations (guided vs. non-guided) showed a high Pearson's correlation coefficient, indicating that they captured the same underlying construct. Similarly, the Jigsaw[2] team adopted a labeling schema that is subjective to individual

[2]https://jigsaw.google.com/the-current/toxicity/.

interpretation [24]. The team found that more annotators could agree on the toxicity of comments based on a more generic guideline. On the contrary, Fortuna et al. [31, 33] suggested using hierarchical multi-class annotation schemes.

The effect of annotators' implicit biases on quality of annotations. Waseem [108] studied the influence of an annotator's expertise on toxicity labeling. They formed two groups of annotators—experts and amateurs—and found a low agreement within these groups across all labels on the W&H dataset. They also found that the amateurs were more likely to assign a toxic label to the sample, leading to a loss in downstream performance. The ratings by the amateurs were also closer to the W&H dataset's crowdsourced annotations. *We need to investigate the impact of the expertise and sensitivity of annotators on the final labels obtained.*

Al Kuwatly et al. [2] investigated the effect of the annotator's demographics on the quality of resulting classifiers. They used the annotators' demographic information (*gender*, *age*, *education,* and *native English speaker*) present in the Wulczyn personal attacks dataset [115]. They found no differences between the two genders. Whereas, models trained on native English speakers' annotations outperformed the ones for non-native speakers in terms of $F1$ score and sensitivity. Similar differences were observed across both groups for age and education. *While this investigation relied on the per-annotator demographics collected during the annotation process, this information can be tricky to collect and limited to the demographic strata considered during the data creation phase.*

Meanwhile, Wich et al. [111] adopted an unsupervised method to group the behavior of annotators. Annotators and the inter-annotator agreements were converted into a graph with the annotators as nodes and their agreement scores as edge weights. Next, a community detection algorithm [9] was used to group similar annotators. They trained toxicity detection systems on annotations by groups with low inter-rater agreements, which resulted in low-performance on the test datasets labeled by other groups. Similar results were seen for a group with higher inter-rater agreement but low agreement with the other groups, indicating a bias in this group's combined annotation. However, the study did not mention obtaining multiple observations for a group, opening the question of variance in performances due to random model initialization. *An exciting extension to this study would be analyzing the type of biases shown by such anomalous groups and if their ratings should be penalized or contrarily given higher weights to improve the inclusion of the demographic represented by them.*

The need to incorporate disagreement in toxicity modeling. In an intriguing study to account for annotation disagreements, Gordon et al. [39] developed a disagreement deconvolution setup to estimate the lower bounds on model performance accounting for variation in human annotations. They compared the drop in performance against the oracle system for respective models trained on datasets covering subjective tasks like toxicity vs. objective tasks like image detection. Researchers observed that when disagreements are not accounted for (i.e., taking a single label per sample), the performance drop is more for subjective tasks than objective ones. Consider a crowdsourced toxicity sample that receives 5 annotations (3 non-hate, 2 hate). When we go by majority voting, the model will consider predicting the label vector as [1,0]. Meanwhile, the probability distribution it should learn is [0.6,0.4]. Framing subjective tasks via majority labeling fails to capture the latent distribution of variable human understanding of toxicity. *The study crucially points out the importance of accounting for disagreements to project the realistic performance of models when deployed in real-world toxicity detection.*

3.3 Lexical Bias

The keyword-based heuristics in data sampling, coupled with a lack of context and human predilection, can cause toxic speech classifiers to learn spurious lexical correlations. The models produce high toxicity ratings for non-toxic texts due to a disproportionately high presence of certain terms

(or phrases) in the content labeled as toxic. This phenomenon is called *lexical bias*. We refer to terms contributing to lexical bias as ***bias sensitive words*** **(BSW)** [5]. This section highlights mechanism to mitigate lexical bias that can be mainly achieved via (a) correcting data labels and (b) debiasing at the modeling level.

Deciding on the target BSW. While Dixon et al. [24] and Kennedy et al. [49] relied on the manual creation of a list of identity terms, Vaidya et al. [103] used the identity attributes annotated in the Civil comments dataset[3] Borkan et al. [12]. Mozafari et al. [64] used **local mutual information (LMI)** to find the terms highly correlated with the toxic labels. Meanwhile, Badjatiya et al. [5] developed unsupervised methods to find the set of BSW.

Mitigation through data correction and filtering. Dixon et al. [24] and Badjatiya et al. [5] proposed new methods for mitigation of lexical bias. However, Zhou et al. [120] investigated methods that have been effective on other NLU debiasing tasks [54, 99].

- **Length-sensitive upsampling:** Dixon et al. [24] found that toxic samples for an identity term are non-uniformly distributed across different lengths. Subsequently, the non-toxic label was upsampled using statements from Wikipedia articles across the length buckets. The authors also observed an improved **pAUCED (pinned AUC**) value on a **synthetic bias evaluation set (SBET)** containing samples equally distributed across the toxicity labels for each identity term. Moreover, the **FPR (False Positive Rate)** values across the identity terms went down without an increase in the variance of the **FNR (False Negative Rate)** values, indicating a reduction in false-positive bias while not affecting the false-negative bias. *Note that these observations were made on SBET, which employs a static target-attribute pair list. Since this list had a different distribution than the initially debiased data, calculations of bias based on SBET may not capture the changing dynamics/semantics when the samples are debiased.* We note the same in our case study of bias-shift (Appendix B).
- **Knowledge-based generalizations:** Badjatiya et al. [5] developed mitigation techniques based on knowledge-based generalizations. They propose lexical database generalization; replacing the BSW occurrence with an ancestor in the wordnet [62] hypernym-tree (e.g., *black* with *color*). Similar to Dixon et al. [24], Badjatiya et al. [5] did not account for the newly acquired biases in the dataset post-mitigation. We present a case study related to this observation in Appendix B.
- **Data filtering:** Zhou et al. [120] explored two automated data filtering approaches (AFLite [54] and DataMaps [99]) to obtain the set of training samples that will lead to better generalizability and reduce bias as a by-product. The correlation of the toxicity label in the Founta dataset [34] with BSWs reduced with both AFLite and DataMaps, corroborating a reduction in the lexical bias in the resulting filtered datasets.

Mitigation through the debiased training of downstream models. Kennedy et al. [49], Vaidya et al. [103], and Zhou et al. [120] studied the mitigation of lexical bias through model-level debiasing.

- **Regularizing importance scores:** Kennedy et al. [49] reduced the weightage a model assigns to identity terms by regularizing cross-entropy loss (L_{CE}):

$$L = L_{CE} + \alpha \sum_{w \in x \cap S} [\phi(w)]^2, \tag{1}$$

[3]During dataset creation, apart from the toxicity labels, samples were also annotated with the presence of lexical markers of 24 identity attributes.

where x is the set of input sequence words, S is the set of identity terms, $\phi(w)$ is the attention score for term w, and α is the strength of regularization. The BERT models trained on the GHC [48]and Stormfront [22] datasets with regularized loss showed a reduction in FPR on the test datasets. Using Sampling and Occlusion [47] as an explanation algorithm, the authors also validated a reduction of the importance score given by the regularized model to the identity terms, suggesting a decrease in lexical bias.

- **Multi-task learning:** Vaidya et al. [103] used **multi-task learning (MTL)** to learn both the toxicity and identity attribute labels for the Civil comments dataset. This loss L accounts for the cross-entropy loss L_{CE_k} for each identity $k \in [0-9]$ in addition to the loss for the toxicity task L_{CE}:

$$L = \sum_{n=1}^{N} \beta_n [\alpha L_{CE}(\hat{y}_n, y_n) + (1-\alpha) \sum_{k=1}^{9} L_{CE_k}(\hat{y}_n^k, y_n^k)], \tag{2}$$

where β_n is a sample weight, given a higher value for a non-toxic example n with at least one identity attribute present. Evaluation of the MTL model on the test split showed an improvement in terms of: (i) AUC, suggesting a better overall model performance, (ii) **Generalized Mean of Bias AUC (GMB AUC)**, suggesting better bias reduction, and (iii) **Background Positive, Subgroup Negative (BPSN AUC)** and **Subgroup AUC (subAUC)** for each identity term, suggesting a reduction in the false-positive bias while not increasing the false-negative bias. *However, this mitigation method relied on the annotation of identity attributes, which is not easy to extract and scale.*

- **Ensemble-based debiasing:** Zhou et al. [120] explored the ensemble-based debiased training method on the Founta dataset. They trained an ensemble of two models: an SVM classifier and a RoBERTa-based classifier. The idea was to let the naive model learn to predict the toxic label based on the biased features, encouraging the robust model to rely on the other unrelated features. Finally, once the ensemble was trained, the naive model was discarded. The final model showed lower *FPR* values on the BSWs containing samples from the Founta dataset, indicating better debiasing. However, it came at the cost of loss in accuracy and $F1$ score.

4 BIASES BASED ON TARGET OF HARM

Overview. The biases introduced during data collection and annotation directly/indirectly impact the ability of toxicity models to predict toxicity against identity groups such as race, gender, age, and so on. Such unintended biasing of models against identity attributes manifests due to specific lexical cues present in the data that are erroneously correlated with class labels. As depicted in Figure 4, these can be linked to either the presence of *non-toxic terms* in high proportion in *toxicity labels* or to the presence of *abusive terms* in high proportion in *non-toxic labels*. For example, the co-occurrence of terms such as "Muslims" and "Islam" in comments marked as Islamophobic leads the model to correlate Muslims with toxicity spuriously. Meanwhile, the presence of phrases like "f*** the world, be you," even though intended to be non-toxic, ends up being annotated or predicted as toxic due to the previously learned relation of slur terms with toxicity.

Meanwhile, the focus on abusive terms or reclaimed slurs falsely causes the models to classify non-toxic content as toxic. The primary goal of mitigating biases against target groups thus aims to reduce the attention assigned to the presence/absence of specific terms and shift the focus to the overall context in the content under consideration. One way of achieving this is by providing external context as meta-data or incorporation of present world knowledge databases. In current literature, toxicity detection and debiasing employ large language models that are well known to

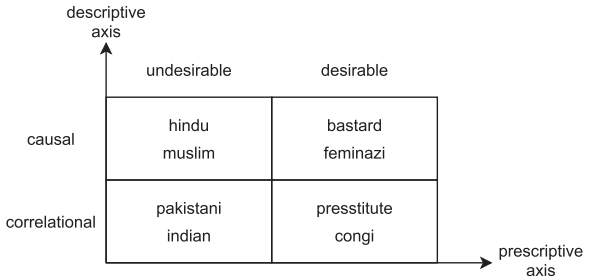


Fig. 4. Ghosh et al. [37] proposed two axes, (i) descriptive and (ii) prescriptive, to segment the associations of the *over-represented* terms for the model under bias evaluation. Here, we consider the cross-section of transferring models trained on English toxicity datasets to toxic phrases in code-mixed Hinglish. The terms in the undesirable quadrants (left column) are those that, despite their frequent presence in toxic classes, are non-toxic/neutral and should not contribute towards toxic label prediction. Meanwhile, the terms in the desirable quadrants (right column) are offensive terms well known in English or colloquial demographic (Indian). For example, while we want the model to correctly predict abusive terms such as bastard or feminazi, or commie to capture toxic intentions, we also want the model to extend to novel lexical terms known to be offensive in the Indian context. For example, "congi" is an offensive connotation to Indian National Congress, a left-leaning political group in India.

be biased in their understanding of identity groups. *Thus, any debiasing of toxicity models is lower bounded by the debiasing of the underlying language frameworks and embeddings employed in the detection models.*

4.1 Racial Bias

Racial bias in machine learning. In their seminal work, Buolamwini and Gebru [14] pointed out that facial recognition systems are biased against dark-skinned faces. Similar issues were observed against Black dialects in automatic speech recognition [101] and the negative association of Black names in popular word embeddings [16]. The issue of biases in word embedding is a much-researched topic in NLP, and its impact extends to downstream tasks of toxicity detection as well. Researchers observed that content posted by Black users or content employing non-white dialects is more likely to be mislabelled as toxic [20]. Thus, already marginalized groups are at a greater risk of being flagged and discriminated against by toxicity detection models. While some of the broader literature around dialect detection is important to applications such as chatbots and automated speech systems, recognizing the dialect based on written content is tricky and contended. The initial work in dialect detection in text revolved around pointing out shortcomings of parsing systems that fail the grammar of diverse dialects; they have been recently adopted to study the impact of dialect in the classification of sentiment and toxicity.

Unfortunately, most literature addressing racial bias primarily focuses on **African-American dialectal English (AAE)**. In our discussion of racial bias in toxicity detection, we will discuss research engaged in evaluating and mitigating racial discrimination against posts attributed to Black users. Several resources have been developed to estimate a poster's race, as these attributes are only sometimes available.

Disclaimer on dialect detection. While some of the broader literature around dialect detection is important to applications such as chatbots and automated speech systems, recognizing the dialect based on written content is tricky and contended. The initial work [25, 26] in dialect detection in text revolved around understanding the changing patterns linguistics; they have been recently adopted to study the impact of dialect in the classification of sentiment and toxicity [85, 120].

However, one has to note that detecting dialect can be a proxy for race and opens up the ethical implication of dialect detection and its application. The implication of the work by References [8] and [44] around automatic dialect detection can thus be problematic unless the context of the user's background is considered. For example, one can easily set up spam accounts on social media to imitate and defame a specific demographic region.

- **Blodgett LM and dataset [8]:** This language model was trained on tweets located in the USA and matched them with the USA census' demographic data for the four largest racial/ethnic groups (namely, those of **non-Hispanic Whites (NHW)**, **non-Hispanic Blacks (NHB)**, Hispanics, and Asians). Out of these, the authors used the language model to filter $1.1M$ (and $14.5M$) Black-oriented (and White-oriented) tweets by likely NHB (and NHW) users. *Using Twitter as a source to train for dialects, Blodgett **LM (Language Model)** could very well employ the same negative word association and stereotyping when predicting dialects, as is the case with toxicity detection models. Utilizing such LMs as black-box for future dialect prediction can exponentially impact the biases against non-white speakers [20]. Methods that use human-annotated dialect labels or do not require ground truth for measuring bias [1] should be favored over dialect-proxy LM-based models.*
- **Pietro dataset [74]:** It is a corpus of $5.4M$ tweets from $4,132$ survey participants who reported their race/ethnicity ($3,184$ NHW and 374 NHB).
- **Huang dataset [44]:** It is a multilingual dataset of tweets combining existing datasets from five languages annotated for toxicity. Four demographic attributes (including race) of the users who posted the comments were also annotated using their Twitter profiles.

In the remainder of the article, we refer to tweets marked as NHB-related (or NHW-related) as Black-oriented (or White-oriented) tweets. We let go of common disputed terminologies such as **Standard-American English (SAE)**, Mainstream US English, and so on, to not promote minority exclusion [80, 120]. Huang et al. [44] noted that racial information encoded in a tweet could introduce downstream harm. They trained models on the English subset of the Huang dataset to predict the author's race solely from the tweet's text. Contrary to the findings of Davidson et al. [20], they found that words like *n*gga* and *b*tch* were more significantly related to the NHW class, suggesting a derogatory use of these terms. *It is an exciting finding, as the Huang dataset is one of the few works not relying on the Blodgett LM for proxy of the user's race.*

Racial information can lead to annotation bias. Black-oriented tweets often get mislabeled as toxic. Sap et al. [85] showed a positive correlation between $p(NHB|tweet)$ and the toxic labels for the Davidson and Founta100k. Similarly, Davidson and Bhattacharya [19] applied **structural topic modeling (STM)** [79] on the Founta100k dataset and identified a latent topic containing terms prevalent in AAE (e.g., *ss, n*gga). to be more likely to be flagged as toxic.

Addressing the biased annotations. Sap et al. [85] and Zhou et al. [120] explored the mitigation of racial bias by addressing annotation bias.

- **Annotator priming:** Sap et al. [85] discovered that annotators were more likely to label a sample as "non-toxic" when primed with the dialect or race of the author. However, it must be noted that annotator priming is a tricky task. Sometimes, it can nurse the annotators' implicit human biases instead of suppressing them.
- **Dialect aware label correction**[4]**:** Zhou et al. [120] utilized the few-shot capabilities of the GPT-3 [13] by supplying a few seed examples [92] for Black-oriented to White-oriented dialect conversion. They used the Founta100k dataset and the Blodgett LM to find the *two *-oriented tweet divisions*. A Black-oriented sample is marked as non-toxic if its White

[4]Zhou et al. [120] cautioned against a real-world application of this method due to limitations of GPT-3.

counterpart is labeled non-toxic. The language model trained on the relabeled dataset led to a lower false positive rate on the Black-oriented test samples.

Diagnosing bias in downstream models. Sap et al. [85] showed that a downstream model trained on either Davidson or Founta100k dataset produced: (i) higher FPR values for the black-oriented comments and (ii) higher FNR values for the White-oriented comments of the test-split. Higher positive rates were also observed for Black-oriented tweets from the Blodgett and Pietro datasets. Similarly, Huang et al. [44] observed higher FPED and FNED values by downstream models trained and tested across all languages of the Huang dataset. Meanwhile, Davidson et al. [20] calculated the proportion of Black-oriented (p_{black}) and White-oriented (p_{white}) tweets from the Blodgett LM predicted as toxic by the models trained on respective training set. A t-test on five popular Twitter toxic speech datasets revealed a significant disparity against the Black-oriented tweets. This disparity reduced (yet persisted) when the t-test was repeated conditioning on the presence of identity terms. The authors noted that while this prevailing disparity could have been due to other terms not conditioned upon, it was also possibly the result of the model correlated other subtler details of AAE with the toxicity.

Mitigation through the debiased training of downstream models. Mozafari et al. [64], Xia et al. [116], Zhou et al. [120] explored mitigation of racial bias through model debiasing.

- **Comment re-weighting in loss:** Following Reference [90], Mozafari et al. [64] defined a bias score s_j^c for each label class c and bigram term t_j as:

$$s_j^c = \frac{\sum_{i=1}^{n} I_{t_j \in x_i} I_{y_i=c}(1+\alpha_i)}{\sum_{i=1}^{n}(1+\alpha_i)}, \tag{3}$$

  where α_i is the weight of the ith sample, and I is the indicator function. These sample weights α were learned by solving the following optimization:

$$min\left(\sum_{j=1}^{|V|} max_c\left(s_j^c\right) + \lambda||\alpha||_2\right), \tag{4}$$

  where V is the set of bigrams in the training data, and λ is a hyperparameter. Though the debiasing was applied to the complete vocabulary, mitigation only in racial bias was evaluated. Reduction in bias against the Black-oriented tweets across all toxicity labels was observed for models trained on the W&H and Davidson datasets.
- **Adversarial training:** Xia et al. [116] used an adversarial training [53] on the Founta100k dataset. One classification head (C) predicted the toxicity label, while the adversary head (D) predicted the protected attribute $p(NHB|tweet)$. This method was shown to be effective when lexical cues between the two attributes are closely related (e.g., terms like n^*gga and b^*tch correlate with both the AAE dialect and toxicity labels [20]). The model showed a reduction in FPR and an increase in macro-$F1$ scores.
- **Ensemble-based debiasing:** Similar to lexical bias (see Section 3.3), Zhou et al. [120] utilized LearnedMixIn for mitigating racial bias on the Founta dataset. They used the four $p(*|tweet)$ predictions from the Blodgett LM as features for the naive model. A drop of FPR on the Black-oriented comments was observed, along with a drop in macro-F1. *However, an insignificant drop in disparity against the Black-oriented tweets was observed when tested on the Pietro dataset, suggesting the inability of mitigation methods on out-of-domain data.*

4.2 Gender Bias

A note on gender in NLP. Bolukbasi et al. [10] pointed out the gendered bias in word embeddings considers a pair of masculine or feminine attributes and job occupations. Since then, multiple studies have evaluated gender bias [96, 104] pinned on the binary gender identity. Only a few studies have explored this phenomenon from a non-binary perspective [118]. Talking specifically of toxicity detection systems, Jigsaw[5] is probably the only dataset that considers more inclusive and fine-grained labels for gender and sexual orientation as the target of toxicity. Unfortunately, the study of non-binary gender bias is missing from the toxicity literature.

Evaluating gender bias. Waseem [108] showed that a model trained on the W&H dataset led to a performance gain by including the user's gender as a feature [86]. It stems from the fact that abusive gendered terms such as "b*t**," "wh***" are more likely to be observed in samples labelled as toxic. Park et al. [70] compared the FPED and FNED values using an SBET [97] model trained on the W&H and Founta datasets. They found that using pre-trained embeddings improved toxic speech detection performance at the cost of more significant equality difference scores. The authors also observed higher bias values for the W&H dataset than the Founta dataset, attributable to Founta's bias-aware design (Section 3.1).

Mitigation techniques. Based on their observations, Nozza et al. [68] and Park et al. [70] further proposed mitigation techniques for gender bias:

- **Debiased embeddings:** Park et al. [70] applied three strategies for gender bias mitigation on the W&H dataset: (i) using debiased-word2vec embeddings [10], (ii) gender-swap data augmentation (using gender pairs identified by Reference [119]), and (iii) transfer learning by first fine-tuning on a less biased large-scale dataset (Founta [34]). The best debiasing performance was achieved by combining debiased embeddings and gender-swapping, reducing FPED and FNED. Meanwhile, the transfer learning approach led to the highest AUC loss. *We attribute this to incompatible labels between the source and the target datasets (abusive vs. sexism).*
- **Length-sensitive upsampling:** Nozza et al. [68] identified 12 common female identity terms in the Evalita2018 [29] dataset. Next, they upsampled tweets from the W&H dataset to balance the count of occurrences of each identity term across the toxic and non-toxic classes of the training dataset. Upsampling while considering the tweet length ranges (similar to Dixon et al. [24]) led to the best gender debiasing. This debiasing, however, came at the cost of a small AUC drop on the test split. *The W&H dataset is known to contain a host of biases that can affect the resulting upsampled dataset.* [113]

4.3 Intersectional Bias

Kim et al. [51] study the intersection of racial and gender bias for AAE. They used the Founta100k dataset and labeled race using the Blodgett LM (cf., Section 4.1). They also marked the dataset with gender and party (political inclination). The party information is used as a control variable. While tweets belonging to the Black community are more likely to be classified as abusive, Black males are significantly more likely to be classified as hateful. In toxic speech detection, it is essential to evaluate such intersectional biases more carefully.

4.4 Cross-geographic Bias

Ghosh et al. [37] observed that most literature in toxicity detection focuses on the English language; this concentrated attention towards a few geographies creates a *knowledge gap* and can

[5] https://www.kaggle.com/c/jigsaw-unintended-bias-in-toxicity-classification/.

Table 3. A Summary of Debiasing Methods Employed for Different Types of Unintended Biases as Discussed in This Survey

Bias	Debiasing Method
Sampling	Topic proximity [104], Data mixing (augmentation) [4]
Annotation	Disagreement deconvolution [39]
Lexical	Length-sensitive sampling [24], Knowledge-based term generalization [5], Data filtering [120], Regularization [49], Multi-task learning [103], Ensemble modeling [120]
Racial	Annotation priming [85], Label correction [120], Regularization [90], Adversarial training [116], Ensemble modeling [120]
Gender	Length-sensitive sampling [68], Debiased LM [10], Gender Swapping [119], Transfer learning [70]

lead to lexical bias. As a concrete example; they showed that Alphabet's Perspective API[6] gives lower toxicity scores to terms that are considered toxic in Indian context (for example, *presstitute*, a slang combining the terms *press* and *prostitute*) leading to false negative predictions. Conversely, terms such as *muslim* and *hindu* were observed to generate higher toxicity scores even in a non-toxic context, leading to false positives. A set of reproduced terms from their study is provided in Figure 4. Ghosh et al. [37] then proposed a two-step weakly supervised method to detect lexical bias for cross-geocultural harmful content. They carried out this analysis using unlabeled tweets collected from seven countries. In the first step, they identified a set of terms T that were statistically overrepresented in the tweets from a country. Their study separated T along two axes concerning the model under investigation: (i) descriptive axis: correlational vs. causal associations, and (ii) prescriptive axes: desirable vs. undesirable associations. Correlational associations refer to the terms not invoking toxicity in the model, while causal refers to the terms causing higher toxicity predictions. Similarly, desirable associations refer to the terms for which higher toxicity ratings are desirable and vice versa for undesirable. In the second step, to find the correlational associations, they built 33 hand-curated template sentences with varying degrees of toxicity (for example, "You are a <person>," "I dislike <person>"). They replaced the template with toxic entities in T and recorded the change in toxicity score. Any term in T causing the toxicity score to increase while it was not desired to is an instance of spurious correlation. They then employed various mitigation techniques such as deletion, substitution, and balancing and found that these methods did not significantly reduce bias.

4.5 Political Bias

Recently, Wich et al. [112] studied the effect of political bias on hate speech detection. Tweets were collected from political leaders who represented the left-winged, right-winged, and neutral ideologies, with each tweet heuristically marked as non-toxic. Finally, these three politically inclined corpora were used to replace non-toxic comments from an existing corpus [94, 114], resulting in politically biased datasets. It was observed that the right-wing-biased dataset produced statistically lower $F1$ lower than the other two biased datasets—*the work did not explore introducing political bias for the toxic label. We find this assumption unconvincing that politically based tweets were treated as only non-toxic.*

5 MAJOR TAKEAWAYS: A SUMMARY

This section highlights the key takeaways from surveying existing debiasing methods. The summary table of various debiasing methods for the respective bias is provided in Table 3. We also compare physical systems and bias as physical entities. This analogy highlights the drawbacks

[6] https://perspectiveapi.com/.

of existing debiasing methods and motivates a continuing integrated system for mitigating biases.

Inspiration from physical systems. We extend the interpretation that "*energy can neither be created nor destroyed but only transformed from one form to another*" to bias as a physical entity. One can say that as long as humans are bombarded with more information than they can process, some form of cognitive biases will keep existing and evolving as per the social structures of the zeitgeist. It also means that, despite our best efforts, our interactions (online or offline) will be riddled with biases and stereotyping—the same manifests as biases in downstream tasks of NLP. Just like the "*system at rest remains at rest unless acted upon by an external force,*" to consciously and effectively overcome biases, we need to supply our NLP pipelines with external impetus in the form of debiasing evaluations and criteria. However, one-time mitigation and evaluation of bias, as tested in the controlled setting by all the techniques discussed in this survey, will unfortunately not be an accurate (or even valid) indicator of bias reduction as the system evolves. The principle that the "*entropy of a closed system never decreases*" renders our one-time bias mitigation efforts moot. Translating entropy to the close-minded biases we harbor, our biases, once seeded, will not automatically disappear. Instead, bias, as an entity, is expected to reinforce and strengthen with time, i.e., years of systematic discrimination and marginalization leading to increased instances of hate speech (offline and online) and hate crime. Thus, we need a continuously probing setup (a regular supply of external energy) to keep the biases in the NLP pipeline in check. We can only ensure their mitigation in an ever-evolving system of human interactions by evaluating biases at every step of the workflow and analyzing it at regular intervals.

- **Categories of Bias:** As we observe in Figure 1, the bias categories based on the *sources* of harm are not necessarily exclusive. Lexical bias can be a consequence of sampling or annotation bias or both. Similarly, any category of bias based on downstream harms can be an artifact of all three categories based on the source of harm.
- **Sampling Bias:** These observations collectively suggest that the text source and the topics used for sampling have a greater influence on the bias characteristics of the dataset than the sampling strategy. Additionally, OSNs have varying tolerance towards toxic speech [65]. The difference in platform-specific policies can further affect the bias characteristics of the sampled dataset.
- **Lexical Bias:** There should be a balance of sensitive words and phrases such as explicit abuse, identity terms (especially that of minority groups), and topic words across the labels to not fabricate any spurious lexical correlations with the labels. These correlations can disrate the model's ability to capture the context of such terms. A common point of failure of models trained on such data is the conflation of identity disclosures (e.g., "I am gay") with identity attacks (e.g., "I hate all gays"), further promoting disparity among social groups.
- **Annotation Bias:** The current literature for annotating harmful content is spread on the spectrum from strict to loosely defined guidelines. Consequently, we observe that the annotation agreements also vary, with no fixed benchmark for toxic content labeling. The observations and results from the above sections highlight the importance of examining the sensitivity of the annotators towards socio-cultural factors such as dialects and race. To effectively study annotators' behavior across attributes, it is essential to have a fair number of participants.
- **Racial Bias:** Every mitigation strategy suggested to reduce racial bias in existing models assumed that both Black-oriented and White-oriented samples follow the same conditional probability $p(Y|X)$. This assumption is flawed, because using specific terms is socially acceptable for a Black person while unacceptable for someone else. Though the resources from References [8, 44, 74] have aided the investigation of racial bias in toxic speech, none of them

have access to the ground-truth labels for the dialect identities. For example, it is possible that the Blodgett LM spuriously correlates the presence of terms like "*n***gga*" and "*b***tch*" with the tweet being Black-oriented [20]. Such predictions can magnify the racial bias during toxicity detection. Moreover, the assumption that only certain social groups employ specific terms is flawed in today's era. While, on the one hand, it leads to broader adoption and acceptance of cultural terms, on the other hand, such terms are employed by extremist groups and trolls for misappropriation and mocking other cultures. Without knowing the background and intent of the online users, it is difficult to pinpoint what they wish to achieve when adopting non-native terms and dialects. The lack of awareness about dialects and their cultural importance among NLP researchers can itself be a source of bias.

6 DIRECTION FOR FUTURE RESEARCH

This section discusses the common challenges across bias mitigation in toxic speech and their possible solutions. Some of these challenges point to the general area of toxicity detection modeling and has been touched upon in other surveys on toxic speech [17, 117].

- **Cognizance towards side effects.** Owing to resource constraints, the study of bias and its mitigation has focused on reducing only one bias at a time. While bias mitigation is vital for toxic speech detection, it is important to acknowledge its ability to introduce newer biases in the pipeline. As stated earlier, the taxonomy of bias is overlapping in nature. For example, lexical bias can be a source of racial and gendered harm. Meanwhile, the lexical bias could be introduced due to spurious data collection and annotation biases. We observe that researchers often failed to acknowledge this critical aspect while evaluating their proposed mitigation methods (Appendix B). Initial work in the direction of intersectional bias has been led by Kim et al. [51], who analyzed the combined impact of gender and race (Section 4.3). However, the analysis and evaluation of the interplay of various biases on toxicity detection remains an open question.
- **Data collection and annotation.** As observed and discussed in Section 3.1, the source and topic of content can have an overarching impact on the characteristics of the dataset curated. While random sampling is closer in characteristics to the real-world distribution, they are highly skewed towards non-hate, which makes collecting toxic comments hard. Meanwhile, priming for specific topics, hashtags, or users to increase the toxic content introduces unintended biases into the dataset and the modelling pipeline. Recently Rahman et al. [77] proposed an **information retrieval (IR)**-based approach to collecting hate speech from Twitter. Their IR-inspired method increased the coverage of hate compared to existing datasets. Such cross-domain methods can help increase the relevance of the content that should be filtered for labelling. Keeping the biases in check can lead to better generalization. Meanwhile, proposing and standardizing code books for data annotation of toxicity labels can help reduce the variability in labels employed by different datasets. A case study is the large-scale harassment annotation code book [38] that helped label 35*k* instances of various categories of online harassment, including hate speech. When researchers want to propose a new toxicity class, a clear distinction and relation to existing labels should be made. Another exciting area of research can be employing weakly supervised or unsupervised methods of data augmentation [87] and domain adaption [57] to reduce the need for annotations and improve generalizability of toxicity detection models.

  Gap in social and computational understanding of toxicity. It is also essential to highlight the gap between the social and psychological studies of harassment faced by vulnerable groups and its computation analysis by technical researchers and industrialists. Currently,

the two systems operate in silos with little interaction between them. Such practices lead to uninformed people developing toxicity detection pipelines without a nuanced understanding of the social issues. For example, without being cognizant of the social/mental impact of objectifying women, it can be tricky to crawl public forums that capture toxicity against gendered body imaging. Therefore, we need to closely incorporate and synergize social issues and their corresponding data sampling and annotation.

- **Gender is not just binary.** Existing literature in the area of gender debiasing in NLP as well as in toxicity detection has evaluated gender as binary (male vs. female). In their recent work, Dev et al. [23] provided a general overview of how non-binary individuals are at risk of erasure and misgendering at the hands of existing language models. These harms trickle down to the task of toxicity detection as well and, unfortunately, its full extent has not been studied yet. As observed in the existing analysis of annotation and lexical biases, annotators' lack of awareness around gender fluidity can lead to inconsistent labels. More so, the toxicity models predict both "I am a homosexual" and "I hate homosexuals" as toxic due to the presence of the word "homosexual," which has been historically used to detest the LGBTQ+ community and has only recently been reclaimed.
- **Language is not static.** Extending from the previous point, the case of word reclamation is a part of the discussion around the use of static hate lexicons, offensive dictionaries, and static knowledge graphs, which cannot account for the evolving language and the evolving social-cultural aspects [93]. Recently, Qian et al. [76] proposed a prototypical learned model for hate speech classification that aims to capture the evolving hateful content as it develops. Such models need to be extended to debiasing methods as well.
- **Proactive bias mitigation.** Zhou et al. [120] showed that instead of targeting existing bias-ridden datasets, downstream harms could be better mitigated by incorporating mitigation techniques starting in the early stages of the learning pipeline, such as data sampling and annotation. While the mitigation methods proposed for annotation and sampling biases are tied to their respective steps in the machine learning pipeline, the methods for other categories are distributed throughout the pipeline. Additionally, most of the surveyed papers demonstrated bias mitigation as a single-step solution [24, 64]. However, it is essential to be "bias-aware" throughout the learning pipeline [70]. A good reference for this can be Reference [98], which formalized the complete pipeline as a sequence of data transformations and defined the potential sources of harm.
- **Out-of-domain evaluation.** Zhou et al. [120] showed that mitigation techniques displaying encouraging results on in-domain samples failed to reduce disparity when tested on out-of-domain datasets. This is an alarming finding, as most mitigation techniques are tested on in-domain data, putting their generalizability in question. This observation entails introducing diverse benchmarks to standardize existing and future work. For example, a benchmark [107] was recently developed to systematically compare gender-bias mitigation techniques in visual recognition models.
- **Collecting user feedback.** The Jigsaw team utilized user feedback as a key source of bias mitigation[7] for its Perspective API. Setting up of a feedback infrastructure, wherever possible, allows a collection of data that better represents the target population.
- **World beyond English text.** In the majority of literature around toxicity detection and debiasing, we mostly consider English datasets. English, being the most widely available language on the Internet, has most accurate preprocessing tools. This has created a knowledge gap when applying toxicity systems for other languages. Especially since what can be

[7] https://medium.com/jigsaw/unintended-bias-and-names-of-frequently-targeted-groups-8e0b81f80a23.

considered toxic in English-speaking geographies may not be considered toxic in other geographies. The initial study in cross-cultural bias is being led by the work of Ghosh et al. [37]. However, the extensive study of toxicity bias in non-English and code-mixed settings remains non-existent. Additionally, the workaround flagging harmful content has focused majorly on text-based features, as they are easier to collect. Meanwhile, the usage of memes and videos (short clips and long ones) spreading toxic and harmful content has been gaining momentum [50, 72, 73]. We need to study the impact of bias in multi-modal content as well. The combination of the individual as well as unified multi-modal bias is an exciting and open area of research.

7 CONCLUSION

While a reasonable amount of work has been dedicated to unintended bias in toxic speech detection, we observe that its mitigation needs further exploration. Due to the multi-faceted and dynamic nature of online toxicity and its unintended biases, conducting a systematic study helps better understand the scope of the bias mitigation strategies. To our knowledge, no survey approaches this subject, focusing on the proposed methods. Therefore, we filled the gap through this survey. We developed a taxonomy of bias based on their source and target of harm. This categorization enables us to effectively discuss these methods and their drawbacks, challenges, and future work directions. We also draw attention to the need to handle more psychographic biases. For example, toxic speech based on political interests is a known issue on OSNs. However, the study of discrimination based on political leanings in harmful speech detection is still an under-explored avenue [112]. We also conducted a case study to introduce the concept of bias shift due to knowledge-based bias mitigation methods. While certainly not exhaustive, we called attention to a list of common challenges and pitfalls of bias-handling methods for toxic speech detection.

APPENDICES

A EVALUATION METRICS

A.1 Background on Bias Evaluation

This section presents about two popular bias evaluation metrics outside the task of classification. This section briefly discusses two popular bias evaluation metrics outside the classification task.

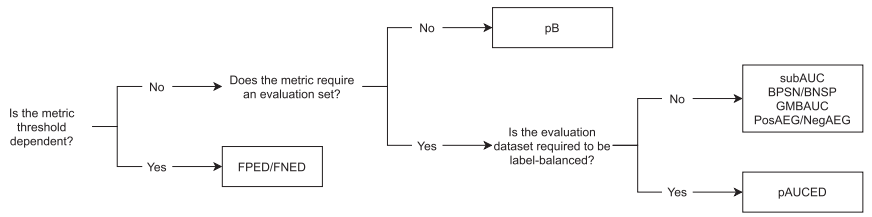


Fig. 5. A taxonomy of bias evaluation metrics popular in toxic speech detection.

Bias evaluation based on psychological tests: Back in 1988 Greenwald et al. [41] developed the **implicit association test (IAT)**[8] as a measure to capture the subconscious biases in humans. It is based on observing the difference in reaction times and accuracy in categorizing two words settings. The first setting relates male with science and female with liberal arts. The second setting reverses the relations. In both settings, the test subjects are tasked to categorize a word towards

[8] https://implicit.harvard.edu/implicit/.

either of these relations. It was observed that most test subjects could categorize words faster and accurately in the first relation (male = science) compared to the second (female = science). *Note, the reliability of IAT is questionable [88], and it can at best be considered only a weak indicator of bias.*[9]

Caliskan et al. [15] extended IAT to develop the **Word Embedding Association Test (WEAT)**. It measures social biases through difference in association strengths in non-contextual word embeddings (e.g., word2vec [61], GloVe [71]). It should be noted that the effectiveness of WEAT as a metric heavily depends on the terms used to seed the algorithm. Later, May et al. [60] extended WEAT for sentence embeddings, naming it the **Sentence Encoding Association Test (SEAT)**. SEAT used templates ("This is a[n] <word>") to insert individual words into the slot "<word>." These templates were designed to convey minimal focus towards the context of the inserted words, helping measure the associations a sentence encoder makes with these inserted terms. Next, it applied WEAT to these synthetic sentences' embeddings instead of word embeddings. In other words, SEAT is a generalization of WEAT on multi-word sequences.

Bias evaluation based on finding bias subspace: Bolukbasi et al. [10] defined the gender bias of a word as the correlation between the projection of the word embedding onto the gender sub-space and the manually annotated bias rating of the word [97]. This gender subspace itself was found by applying **principal component analysis (PCA)** on a set of vectors obtained by taking a difference of words that vary only in the gender context (e.g., $\vec{\text{king}} - \vec{\text{queen}}$) and retaining the top component(s). Liang et al. [55] extended this idea to contextual sentence embeddings. Unlike Bolukbasi et al. [10], to calculate the gender sub-space, they used a large number of sentence pairs, with each sentence in a pair varying just in the gender-specific word. Both the bias evaluation categories described are yet to find a direct utility in the field of bias evaluation in toxic speech classification. One rudimentary exploration for the toxic speech classification task can be the analysis of social biases in fine-tuned word embeddings or sentence encoders after training them for the downstream task dataset.

A.2 Unintended Bias Metrics

This subsection discusses metrics developed especially for bias evaluation of classification models. As described in Figure 5, we create a taxonomy of bias evaluation metrics standard in toxic speech detection literature. We list the metric abbreviations along with other commonly used abbreviations in the article in Table 2. In the following definitions, we assume T as a set of terms $t_1, t_2, \ldots, t_{|T|}$. The metrics defined for a term t can also be calculated for a designated group g among a host of vulnerable groups G.

For a test dataset D, the following definitions apply: (i) let D_t^+ be the positive (toxic) examples containing term t, (ii) let D_t^- be the negative (non-toxic) examples containg term t, (iii) let $D_{\setminus t}^+$ be the positive (toxic) examples not containing the term t, and (iv) let $D_{\setminus t}^-$ be the negative (non-toxic) examples not containing the term t.

- **Error rate equality differences (FPED, FNED):** Dixon et al. [24] introduced the metrics **False Positive Equality Difference (FPED), False Negative Equality Difference (FNED)**, and **pinned AUC Equality Difference (pAUCED)**. FPED and FNED quantify a variation of equality of opportunity. Mathematically, the authors calculated these variations of term-wise error rates (FPR_t and FNR_t) around the error rates (FPR and FNR) of the complete evaluation set:

$$FPED_T = \sum_{t \in T} |FPR - FPR_t|, \tag{5}$$

[9] https://www.apa.org/monitor/2008/07-08/psychometric.

$$FNED_T = \sum_{t \in T} |FNR - FNR_t|. \tag{6}$$

Note that in an ideal case, $FPED_T = FNED_T = 0$. Both FPED and FNED are threshold-dependent and require a classifier that produces binary labels. However, many models produce probability distributions. Dixon et al. [24] and Borkan et al. [12] applied multiple threshold-agnostic metrics for such scenarios.

Usage: In general, FNR and FPR capture the upper bound on misclassification that a model can currently achieve. When employing FNR_t and FPR_t on different identity terms, if the model is biased towards the terms, then there will be higher variance in FNR_t and FPR_t, leading to higher FPED and FPED.

- **Sub-group AUC (subAUC$_t$):** It calculates AUC on $(D_t^+ \cup D_t^-)$, i.e., the examples containing the identity term t.
  Usage: $subAUC_t$ based on specific terms measures the model's ability to correctly predict the toxic and non-toxic classes based on t. A lower value means the model is not able to distinguish the toxic samples from the non-toxic ones containing term t. A higher value, however, does not mean the model is debiased. It can still give non-toxic mentions of term t a high toxicity score, but as long as it gives a toxic mention of t a relatively higher toxicity score than its non-toxic counter part, $subAUC_t$ will be high.
- **Pinned AUC (pAUCED)**: **Area under the receiver operating characteristic curve (AUC-ROC or AUC)** on the complete evaluation set can be insufficient at diagnosing bias, as a low AUC does not help in identifying the bias-ridden terms in T. Similar limitation has been noticed for $subAUC_t$. To overcome these limitations, Dixon et al. [24] developed **pinned AUC ($pAUCED_t$)** for a term t or a subgroup g, which is the AUC measure on an auxiliary dataset pD_t such that $pD_t = s(D_t) \cup s(D)$ and $|s(D_t)| = |s(D)|$, where D_t is the set of comments containing term t in the evaluation set, D is the complete evaluation set, and $s(.)$ is a sampling function. Here, the auxiliary data allows examples from $s(D)$ to be selected at random while "pinning" down the underlying distribution for the samples containing term t, i.e., the distribution of t in pD_t should resemble the overall distribution of t in D as closely as possible. In layman's terms, one can also assume the "pinned" samples to be the "representative" samples capturing the overall distribution of the control variable under consideration. In the toxicity study, these control variables are the identity terms t or the vulnerable group g. However, Borkan et al. [11] showed that the ability of $pAUCED$ to reveal bias is highly dependent on the distribution of labels between the identity terms. Dixon et al. [24] avoided this drawback by generating a **synthetic bias evaluation set (SBET)** with balanced label distribution. However, generating a SBET can be a tedious task [5].
  Usage: When we have multiple subgroups $g \in G$, we can employ pinned AUC to quantify and rank the level of unintended bias of the model w.r.t. various subgroups. The group with lowest pinned AUC is carrying the highest bias.
- **Background Positive Subgroup Negative AUC (BPSN), and Background Negative Subgroup Positive AUC (BNSP)**: $BPSN$ calculates AUC on test set where non-toxic samples contain the bias term t, while toxic samples that do not, i.e., $(D_t^- \cup D_{\setminus t}^+)$. Meanwhile, in BNSP AUC, we do the reverse, selecting the toxic samples that mention a term t and non-toxic ones that do not, i.e., $(D_t^+ \cup D_{\setminus t}^-)$. The former leads to reduction in FPR, while the latter leads to reducing in FNR.
  Usage: A lower BPSN means the model confuses non-toxic mentions of the identity term (t) with toxicity examples that do not contain the term. In other words, the model assumes the non-toxic samples containing t to be close to the toxicity class in general. For example,

when the term "nigga" is used in a post, "I feel for you, my nigga," the model focuses on the high correlation between "nigga" and toxicity instead of the neutral context.
However, a lower BNSP means the model confuses toxic samples that contain t with non-toxic samples that do not contain the term. Here, even the toxic usage of t is assumed to be closer to the non-toxicity class. For example, the sentence "Whites are superior to the rest" may not receive high toxicity as the identity term "White" usually carries low negative association without context. One can blame sampling and annotation (Sections 3.1 and 3.2) biases for this.

- **Generalized Means of Bias AUC (GMBAUC)**: To obtain a mean bias score across different bias AUC metrics, Jigsaw[10] introduced a generalized mean of the bias AUC (or GMBAUC) as:

$$GMBAUC_{p,T} = (0.25 \times AUC) + \sum_{a=1}^{3} 0.25 \times \left(\frac{1}{|T|}\sum_{t=1}^{|T|} m_{a,t}^{p}\right)^{\frac{1}{p}}, \tag{7}$$

where AUC is the overall ROC AUC score, $m_{a,t}$ is the ath AUC-based metric calculated for term t, and p is the power of the mean function. Inclusion of overall AUC helps in capturing the downstream performance.
Usage: GMBAUC is, essentially, an average of the subgroup AUCs that helps to easily compare different debiasing settings by having to follow only one metric. A lower GMB AUC like subgroup-specific AUC is a good indicator of unintended bias in the toxicity detection pipeline against one or more groups.

- **Pinned Bias (pB):** In a work parallel to Reference [12], Badjatiya et al. [5] introduced a family of bias evaluation metrics, pinned Bias (pB):

$$pB_T = \sum_{t \in T} \frac{|p(\text{“}toxic\text{”}|t) - \phi|}{|T|}, \tag{8}$$

where (i) $p(\text{“}toxic\text{”}|t)$ is the prediction probability of the toxicity label for a sentence containing only term t, (ii) ϕ is the pinned/threshold value that differs for different members of the metric family. For example, to penalize $p(\text{“}toxic\text{”}|t)$ values above 0.5, $\phi = min(p(\text{“}toxic\text{”}|t), 0.5)$ can be used. Here, the pinning is on the threshold probability that depicts how much toxicity can be allowed for a term. The limitation of above formulation is that ϕ is constant for all terms in T, while in the real world the thresholds are dynamic.
Usage: pB can be used in tandem or even as a replacement for pinned AUC. While pAUCED requires the creation of a balanced auxiliary dataset for each term $t \in BSW$, pB utilizes the same dataset and is easier to measure.

B CASE STUDY ON KNOWLEDGE-DRIFT

Lexical debiasing. To overcome lexical bias (Section 3.3), Badjatiya et al. [5] employed a knowledge-based generalization method. It involved the replacement of the BSWs in the training dataset with an ancestor from the WordNet [62] hypernym-tree.

Knowledge-drift hypothesis. Motivated by the energy-bias analogy (refer to Section 5), we conjecture that replacing all occurrences $w \in BSW$ with its wordnet ancestor $a \in A$ will shift the bias from w to a. While generalization can be true for all the debiasing methods we discussed in this survey, for this experiment, we focus on the wordnet-based substitution method for lexical debiasing.

[10] https://www.kaggle.com/c/jigsaw-unintended-bias-in-toxicity-classification/.

Limited evaluation. Post substitution of W with A, the authors employed the BSW on the original W to evaluate bias. There is no discussion on evaluating the bias on terms in A, which is now a substitute for W.

Experimental setup. All experiments were carried out on a Ubuntu 18.04.5 LTS system with 126 G RAM and 32 G Tesla V100. We apply debiasing method on the W&H dataset [109] to verify the hypothesis and compare by training a BERTweet [67] classifier.

Table 4. (a): BSWs w with Their Corresponding Selected Generalizations a According to the Wordnet-3 Scheme [5]. (b): pB Values for the Set of BSWs W and A for (i) the Model Obtained on Original W&H Dataset M_{bias} and (ii) the Model Obtained after Lexical Database Generalization M_{gen}

(a)

$w \in W$	$a \in A$
muslim, prophet, woman, christian, girl, terrorist, slave, man, child, driver	being
feminist, civilian, liar, comedian, god	someone
hate, slavery, hatred, want, truth, freedom	state

(b)

Metric	M_{bias}	M_{gen}
pB_W	0.027	0
pB_A	0.002	0.016

Observation. Table 4(a) lists the replaced words and their respective generalizations. The observations in Table 4(b) indicate a shift of bias from source to target words. Since the terms in the set A are more likely to be present in non-toxic comments, this shift in bias can also be detrimental to the non-toxical class. While this is a proof of concept case study, it is expected that the other mitigation techniques by Badjatiya et al. [5] based on knowledge-based generalizations suffer from a similar shift of bias instead of actually debiasing the dataset.

Resolving bias shift. Note that this shift of lexical bias is towards a more general set of terms A, where debiasing on this generalized dataset through upsampling [24] directly from an OSN (such as Twitter) seems like a feasible next step. For example, newly scraped comments containing terms such as *muslim* can not be directly added to the training dataset. It is because the number of toxic comments can be expected to be higher. However, if *muslim* is first generalized to *being*, then randomly scraping new comments with this new keyword can lead to a significantly lower toxicity ratio.

C REPRODUCIBILITY OF DEBIASING METHODS

Motivation. In this section, we reproduce some of the bias mitigation methods described in the respective sections of bias. We mainly extend upon the results reported by Zhou et al. [120] and Park et al. [70] to incorporate W&H [109] and Davidson [21] datasets along with Founta [34]. By reproducing these methods, we can comment not just on the theoretical discussion of the debiasing techniques but also on their engineering feasibility. We hope to give the readers a glimpse of various settings (dataset, embedding, evaluation combination) for the debiasing methods presented in this survey.

Experimental Setup. All experiments were carried out on a Ubuntu 18.04.5 LTS system with 126 G RAM and 32 G Tesla V100. The hyper-parameters and epochs were kept intact as provided in the respective methodology.

C.1 Racial Bias

For studying racial bias, we reproduce the results reported in Zhou et al. [120]. As observed in Table 5(c) for the large-scale dataset (like Founta [34]), data filtering methods like Aflite that

Table 5. Reproduced Results for Racial Debiasing as Experimented by Zhou et al. [120]

(a)

Method/Result	W FPR	B FPR	W %	B %
Random	0.072	0.133	0.267	0.481
Original	0.025	0.024	0.162	0.380
LearnedMixIn	0.028	0.024	0.178	0.405

(b)

Method/Result	W FPR	B FPR	W %	B %
Random	0.029	0.022	0.045	0.033
Original	0.032	0.009	0.063	0.015
LearnedMixIn	0.193	0.308	0.194	0.313

(c)

Method/Result	W FPR	B FPR	W %	B %
Random	0.017	0.058	0.303	0.719
Original	0.040	0.115	0.300	0.725
Datamap-Easy	0.018	0.053	0.302	0.717
Datamap-Ambi	0.020	0.077	0.307	0.725
Datamap-Hard	0.020	0.073	0.287	0.716
Aflite	0.017	0.058	0.303	0.719
LearnedMixIn	0.0275	0.075	0.281	0.694

(a) Results on W&H [109], (b) Results on Davidson [21], (c) Results on Founta [34].

eventually operate on only 33% of the dataset provide a comparable drop (less is more) in *FPR*. Meanwhile, the LearnedMixIn method [18] that adopts model regularization provides the smallest percentage of Black tweets samples wrongly labelled as toxic. For W&H and Davidson datasets, applying data filtering and utilizing only 33% of the dataset would leave us only $\approx 5k$ and $\approx 8k$ tweets, respectively. Hence, we did not test data filtering on small-scale datasets and only reported unexpected and LearnMixIn results. Debiasing in both datasets via LearnMixIn (Tables 5(a) and 5(b)) is not better than the original dataset composition. This we assume could be happening due to the small volume of samples and the presence of other biases (lexical, topical) in the datasets.

C.2 Gender Bias

Here, we reproduce the results from Park et al. [70] by testing the combination of embedding and model against an exhaustive set of gender debiasing techniques. For the transfer learning, we test on Waseem and Davidson using Founta for training. In line with existing literature, we observe in Table 6 that pre-trained word embedding, on average, gives higher FPED and FNED due to their intrinsic gender bias. Similarly, as observed from Table 7, transfer learning leads to a higher F1 score, but it comes at the cost of increasing FPED and FNED. A combination of debiased word2vec, gender-swapping, and transfer learning, while a more complex set, seems to be the best tradeoff between reducing F1 and increasing FPED-FNED values. However, no one setting is best across datasets.

Table 6. Reproduced Results for Gender Debiasing Employing CNN, GRU, and α-GRU on Three Different Embeddings: *R*: Randomly Initialized, *F*: Fasttext, *W*: word2vec

(a)

Method	Emd.	FPR	FNR	FPED	FNED	F1
CNN	R	0.039	0.331	0.067	0.586	0.885
	F	0.060	0.213	0.097	0.384	0.902
	W	0.053	0.247	0.088	0.445	0.899
GRU	R	0.076	0.258	0.134	0.439	0.879
	F	0.071	0.277	0.120	0.477	0.877
	W	0.082	0.227	0.141	0.400	0.882
α-GRU	R	0.043	0.243	0.073	0.426	0.907
	F	0.080	0.228	0.142	0.414	0.885
	W	0.080	0.202	0.142	0.365	0.892

(b)

Method	Emd.	FPR	FNR	FPED	FNED	F1
CNN	R	0.010	0.842	0.019	1.561	0.917
	F	0.013	0.762	0.025	1.415	0.926
	W	0.008	0.768	0.016	1.439	0.928
GRU	R	0.043	0.654	0.082	1.222	0.916
	F	0.059	0.624	0.108	1.160	0.908
	W	0.035	0.735	0.065	1.370	0.914
α-GRU	R	0.037	0.750	0.069	1.372	0.912
	F	0.041	0.695	0.074	1.274	0.914
	W	0.047	0.683	0.087	1.256	0.910

(c)

Method	Emd.	FPR	FNR	FPED	FNED	F1
CNN	R	0.04	0.327	0.071	0.586	0.883
	F	0.048	0.247	0.077	0.449	0.902
	W	0.044	0.262	0.069	0.487	0.901
GRU	R	0.143	0.325	0.254	0.560	0.786
	F	0.070	0.125	0.125	0.216	0.909
	W	0.097	0.119	0.174	0.203	0.89
α-GRU	R	0.081	0.232	0.142	0.422	0.883
	F	0.098	0.183	0.170	0.327	0.884
	W	0.081	0.202	0.138	0.357	0.891

(a) Results on W&H [109], (b) Results on Davidson [21], (c) Results on Founta [34].

Table 7. Reproduced Results for Gender Debiasing Employing CNN, GRU, and α-GRU on a Combination of Three Debiasing Techniques: *DE*: Debiased word2vec, *GS*: Gender Swapping, and *FT*: Transfer Learning by Training on Founta [34]

(a)

Method	DE	GS	FT	FPR	FNR	FPED	FNED	F1
CNN	✓	-	-	0.0294	0.316	0.045	0.567	0.897
	-	✓	-	0.047	0.297	0.080	0.555	0.890
	-	-	✓	0.054	0.331	0.090	0.600	0.875
	✓	✓	-	0.054	0.267	0.087	0.491	0.893
	-	✓	✓	0.118	0.909	0.218	1.422	0.644
	✓	-	✓	0.124	0.871	0.223	1.365	0.656
	✓	✓	✓	0.118	0.909	0.218	1.422	0.644
GRU	✓	-	-	0.070	0.297	0.123	0.515	0.872
	-	✓	-	0.076	0.247	0.130	0.427	0.882
	-	-	✓	0.095	0.427	0.166	0.723	0.818
	✓	✓	-	0.048	0.342	0.084	0.581	0.874
	-	✓	✓	0.134	0.816	0.252	1.27	0.668
	✓	-	✓	0.064	0.712	0.118	1.185	0.746
	✓	✓	✓	0.065	0.708	0.119	1.177	0.746
α-GRU	✓	-	-	0.065	0.297	0.115	0.525	0.877
	-	✓	-	0.075	0.255	0.130	0.456	0.881
	-	-	✓	0.061	0.278	0.107	0.487	0.885
	✓	✓	-	0.097	0.232	0.169	0.400	0.872
	-	✓	✓	0.071	0.247	0.132	0.426	0.886
	✓	-	✓	0.085	0.243	0.148	0.445	0.877
	✓	✓	✓	0.092	0.247	0.155	0.437	0.871

(b)

Method	DE	GS	FT	FPR	FNR	FPED	FNED	F1
CNN	✓	-	-	0.446	0.537	0.791	1.0	0.658
	-	✓	-	0.008	0.793	0.016	1.476	0.925
	-	-	✓	0.037	0.098	0.056	0.149	0.940
	✓	✓	-	0.008	0.860	0.015	1.567	0.917
	-	✓	✓	0.085	0.927	0.148	1.707	0.867
	✓	-	✓	0.235	0.774	0.424	1.446	0.792
	✓	✓	✓	0.129	0.860	0.224	1.580	0.849
GRU	✓	-	-	0.0415	0.685	0.075	1.278	0.914
	-	✓	-	0.035	0.735	0.065	1.370	0.914
	-	-	✓	0.491	0.501	0.712	1.0	0.679
	✓	✓	-	0.0468	0.716	0.083	1.29	0.908
	-	✓	✓	0.009	1.0	0.014	1.821	0.898
	✓	-	✓	0.003	1.0	0.004	1.846	0.901
	✓	✓	✓	0.006	0.932	0.011	1.698	0.909
α-GRU	✓	-	-	0.034	0.762	0.0618	1.409	0.912
	-	✓	-	0.031	0.738	0.057	1.366	0.916
	-	-	✓	0.079	0.120	0.123	0.182	0.906
	✓	✓	-	0.035	0.781	0.065	1.402	0.909
	-	✓	✓	0.041	0.762	0.075	1.378	0.907
	✓	-	✓	0.038	0.787	0.068	1.439	0.907
	✓	✓	✓	0.014	0.860	0.024	1.555	0.914

✓marks the combination of debiasing used. (a) Results on W&H [109], (b) Results on Davidson [21].

REFERENCES

[1] Osman Aka, Ken Burke, Alex Bauerle, Christina Greer, and Margaret Mitchell. 2021. Measuring model biases in the absence of ground truth. In *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society (AIES'21)*. Association for Computing Machinery, New York, NY, 327–335. DOI: https://doi.org/10.1145/3461702.3462557

[2] Hala Al Kuwatly, Maximilian Wich, and Georg Groh. 2020. Identifying and measuring annotator bias based on annotators' demographic characteristics. In *Proceedings of the 4th Workshop on Online Abuse and Harms*. 184–190.

[3] Nuha Albadi, Maram Kurdi, and Shivakant Mishra. 2018. Are they our brothers? Analysis and detection of religious hate speech in the Arabic Twittersphere. In *Proceedings of the IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM'18)*. IEEE, 69–76.

[4] Aymé Arango, Jorge Pérez, and Barbara Poblete. 2019. Hate speech detection is not as easy as you may think: A closer look at model validation. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 45–54.

[5] Pinkesh Badjatiya, Manish Gupta, and Vasudeva Varma. 2019. Stereotypical bias removal for hate speech detection task using knowledge-based generalizations. In *Proceedings of the World Wide Web Conference*. 49–59.

[6] Ricardo Baeza-Yates. 2020. Bias in search and recommender systems. In *Proceedings of the 14th ACM Conference on Recommender Systems (RecSys'20)*. Association for Computing Machinery, New York, NY, 2. DOI: https://doi.org/10.1145/3383313.3418435

[7] Su Lin Blodgett, Solon Barocas, Hal Daumé III, and Hanna Wallach. 2020. Language (technology) is power: A critical survey of "Bias" in NLP. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. 5454–5476.

[8] Su Lin Blodgett, Lisa Green, and Brendan O'Connor. 2016. Demographic dialectal variation in social media: A case study of African-American English. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 1119–1130. DOI: https://doi.org/10.18653/v1/D16-1120

[9] Vincent D. Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. 2008. Fast unfolding of communities in large networks. *J. Statist. Mechan.: Theor. Experim.* 2008, 10 (2008), P10008.

[10] Tolga Bolukbasi, Kai-Wei Chang, James Zou, Venkatesh Saligrama, and Adam Kalai. 2016. Man is to computer programmer as woman is to homemaker? Debiasing word embeddings. In *Proceedings of the 30th International Conference on Neural Information Processing Systems (NIPS'16)*. Curran Associates Inc., Red Hook, NY, 4356–4364.

[11] Daniel Borkan, Lucas Dixon, John Li, Jeffrey Sorensen, Nithum Thain, and Lucy Vasserman. 2019. Limitations of pinned AUC for measuring unintended bias. *arXiv preprint arXiv:1903.02088* (2019).

[12] Daniel Borkan, Lucas Dixon, Jeffrey Sorensen, Nithum Thain, and Lucy Vasserman. 2019. Nuanced metrics for measuring unintended bias with real data for text classification. In *Proceedings of the World Wide Web Conference*. 491–500.
[13] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell et al. 2020. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165* (2020).
[14] Joy Buolamwini and Timnit Gebru. 2018. Gender shades: Intersectional accuracy disparities in commercial gender classification. In *Proceedings of the 1st Conference on Fairness, Accountability and Transparency (Proceedings of Machine Learning Research,* Vol. 81), Sorelle A. Friedler and Christo Wilson (Eds.). PMLR, 77–91. Retrieved from: https://proceedings.mlr.press/v81/buolamwini18a.html.
[15] Aylin Caliskan, Joanna J. Bryson, and Arvind Narayanan. 2017. Semantics derived automatically from language corpora contain human-like biases. *Science* 356, 6334 (2017), 183–186.
[16] Aylin Caliskan, Joanna J. Bryson, and Arvind Narayanan. 2017. Semantics derived automatically from language corpora contain human-like biases. *Science* 356, 6334 (2017), 183–186. DOI: https://doi.org/10.1126/science.aal4230
[17] Tanmoy Chakraborty and Sarah Masud. 2022. Nipping in the bud: Detection, diffusion and mitigation of hate speech on social media. arXiv:2201.00961 [cs.SI].
[18] Christopher Clark, Mark Yatskar, and Luke Zettlemoyer. 2019. Don't take the easy way out: Ensemble based methods for avoiding known dataset biases. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP'19)*. Association for Computational Linguistics, 4069–4082. DOI: https://doi.org/10.18653/v1/D19-1418
[19] Thomas Davidson and Debasmita Bhattacharya. 2020. Examining racial bias in an online abuse corpus with structural topic modeling. *arXiv preprint arXiv:2005.13041* (2020).
[20] Thomas Davidson, Debasmita Bhattacharya, and Ingmar Weber. 2019. Racial bias in hate speech and abusive language detection datasets. In *Proceedings of the 3rd Workshop on Abusive Language Online*. Association for Computational Linguistics, 25–35. DOI: https://doi.org/10.18653/v1/W19-3504
[21] Thomas Davidson, Dana Warmsley, Michael Macy, and Ingmar Weber. 2017. Automated hate speech detection and the problem of offensive language. In *Proceedings of the International AAAI Conference on Web and Social Media*.
[22] Ona de Gibert, Naiara Perez, Aitor García-Pablos, and Montse Cuadros. 2018. Hate speech dataset from a white supremacy forum. *arXiv preprint arXiv:1809.04444* (2018).
[23] Sunipa Dev, Masoud Monajatipoor, Anaelia Ovalle, Arjun Subramonian, Jeff Phillips, and Kai-Wei Chang. 2021. Harms of gender exclusivity and challenges in non-binary representation in language technologies. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 1968–1994. DOI: https://doi.org/10.18653/v1/2021.emnlp-main.150
[24] Lucas Dixon, John Li, Jeffrey Sorensen, Nithum Thain, and Lucy Vasserman. 2018. Measuring and mitigating unintended bias in text classification. In *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*. 67–73.
[25] Jacob Eisenstein. 2017. *Identifying Regional Dialects in On-line Social Media*. John Wiley & Sons, Ltd, 368–383. DOI: https://doi.org/10.1002/9781118827628.ch21
[26] Jacob Eisenstein, Noah A. Smith, and Eric P. Xing. 2011. Discovering sociolinguistic associations with structured sparsity. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, 1365–1374. Retrieved from: https://aclanthology.org/P11-1137.
[27] Mai ElSherief, Caleb Ziems, David Muchlinski, Vaishnavi Anupindi, Jordyn Seybolt, Munmun De Choudhury, and Diyi Yang. 2021. Latent hatred: A benchmark for understanding implicit hate speech. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 345–363. DOI: https://doi.org/10.18653/v1/2021.emnlp-main.29
[28] Christian Ezeibe. 2020. Hate speech and election violence in Nigeria. *J. Asian Afric. Stud.* 56, 4 (2020), 0021909620951208.
[29] Elisabetta Fersini, Debora Nozza, and Paolo Rosso. 2018. Overview of the Evalita 2018 task on automatic misogyny identification (AMI). *EVALITA Eval. NLP Speech Tools Ital.* 12 (2018), 59.
[30] Agneta Fischer, Eran Halperin, Daphna Canetti, and Alba Jasini. 2018. Why we hate. *Emot. Rev.* 10, 4 (2018), 309–320.
[31] Paula Fortuna, Joao Rocha da Silva, Leo Wanner, Sérgio Nunes et al. 2019. A hierarchically-labeled Portuguese hate speech dataset. In *Proceedings of the 3rd Workshop on Abusive Language Online*. 94–104.
[32] Paula Fortuna and Sérgio Nunes. 2018. A survey on automatic detection of hate speech in text. *ACM Comput. Surv.* 51, 4 (2018), 1–30.
[33] Paula Fortuna, Juan Soler, and Leo Wanner. 2020. Toxic, hateful, offensive or abusive? What are we really classifying? An empirical analysis of hate speech datasets. In *Proceedings of the 12th Language Resources and Evaluation Conference*. 6786–6794.

[34] Antigoni Founta, Constantinos Djouvas, Despoina Chatzakou, Ilias Leontiadis, Jeremy Blackburn, Gianluca Stringhini, Athena Vakali, Michael Sirivianos, and Nicolas Kourtellis. 2018. Large scale crowdsourcing and characterization of Twitter abusive behavior. In *Proceedings of the International AAAI Conference on Web and Social Media*.

[35] Matt Gardner, Jonathan Berant, Hannaneh Hajishirzi, Alon Talmor, and Sewon Min. 2019. On making reading comprehension more comprehensive. In *Proceedings of the 2nd Workshop on Machine Reading for Question Answering*. 105–112.

[36] Ismael Garrido-Muñoz, Arturo Montejo-Ráez, Fernando Martínez-Santiago, and L. Alfonso Ureña-López. 2021. A survey on bias in deep NLP. *Appl. Sci.* 11, 7 (Apr. 2021), 3184. DOI: https://doi.org/10.3390/app11073184

[37] Sayan Ghosh, Dylan Baker, David Jurgens, and Vinodkumar Prabhakaran. 2021. Detecting cross-geographic biases in toxicity modeling on social media. In *Proceedings of the 7th Workshop on Noisy User-generated Text (W-NUT'21)*. Association for Computational Linguistics, 313–328. DOI: https://doi.org/10.18653/v1/2021.wnut-1.35

[38] Jennifer Golbeck, Zahra Ashktorab, Rashad O. Banjo, Alexandra Berlinger, Siddharth Bhagwan, Cody Buntain, Paul Cheakalos, Alicia A. Geller, Quint Gergory, Rajesh Kumar Gnanasekaran, Raja Rajan Gunasekaran, Kelly M. Hoffman, Jenny Hottle, Vichita Jienjitlert, Shivika Khare, Ryan Lau, Marianna J. Martindale, Shalmali Naik, Heather L. Nixon, Piyush Ramachandran, Kristine M. Rogers, Lisa Rogers, Meghna Sardana Sarin, Gaurav Shahane, Jayanee Thanki, Priyanka Vengataraman, Zijian Wan, and Derek Michael Wu. 2017. A large labeled corpus for online harassment research. In *Proceedings of the ACM Web Science Conference (WebSci'17)*. Association for Computing Machinery, New York, NY, 229–233. DOI: https://doi.org/10.1145/3091478.3091509

[39] Mitchell L. Gordon, Kaitlyn Zhou, Kayur Patel, Tatsunori Hashimoto, and Michael S. Bernstein. 2021. The disagreement deconvolution: Bringing machine learning performance metrics in line with reality. In *Proceedings of the CHI Conference on Human Factors in Computing Systems (CHI'21)*. Association for Computing Machinery, New York, NY. DOI: https://doi.org/10.1145/3411764.3445423

[40] Leon Graumas, Roy David, and Tommaso Caselli. 2019. Twitter-based polarised embeddings for abusive language detection. In *Proceedings of the 8th International Conference on Affective Computing and Intelligent Interaction Workshops and Demos (ACIIW'19)*. IEEE, 1–7.

[41] Anthony G. Greenwald, Debbie E. McGhee, and Jordan L. K. Schwartz. 1998. Measuring individual differences in implicit cognition: The implicit association test. *J. Personal. Soc. Psychol.* 74, 6 (1998), 1464–1480. DOI: https://doi.org/10.1037/0022-3514.74.6.1464

[42] Thomas Hartvigsen, Saadia Gabriel, Hamid Palangi, Maarten Sap, Dipankar Ray, and Ece Kamar. 2022. ToxiGen: A large-scale machine-generated dataset for adversarial and implicit hate speech detection. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, 3309–3326. DOI: https://doi.org/10.18653/v1/2022.acl-long.234

[43] Martie G. Haselton, Daniel Nettle, and Paul W. Andrews. 2015. The evolution of cognitive bias. In *The Handbook of Evolutionary Psychology*. John Wiley & Sons, Inc., 724–746. DOI: https://doi.org/10.1002/9780470939376.ch25

[44] Xiaolei Huang, Linzi Xing, Franck Dernoncourt, and Michael J. Paul. 2020. Multilingual Twitter corpus and baselines for evaluating demographic bias in hate speech recognition. *arXiv preprint arXiv:2002.10361* (2020).

[45] Ferenc Huszár, Sofia Ira Ktena, Conor O'Brien, Luca Belli, Andrew Schlaikjer, and Moritz Hardt. 2021. Algorithmic amplification of politics on Twitter. *Proc. Nat. Acad. Sci.* 119, 1 (Dec. 2021), e2025334119. DOI: https://doi.org/10.1073/pnas.2025334119

[46] Òscar Garibo i Orts. 2019. Multilingual detection of hate speech against immigrants and women in Twitter at SemEval-2019 task 5: Frequency analysis interpolation for hate in speech detection. In *Proceedings of the 13th International Workshop on Semantic Evaluation*. 460–463.

[47] Xisen Jin, Zhongyu Wei, Junyi Du, Xiangyang Xue, and Xiang Ren. 2020. Towards hierarchical importance attribution: Explaining compositional semantics for neural sequence models. In *Proceedings of the International Conference on Learning Representations*. Retrieved from: https://openreview.net/forum?id=BkxRRkSKwr.

[48] Brendan Kennedy, Mohammad Atari, Aida Mostafazadeh Davani, Leigh Yeh, Ali Omrani, Yehsong Kim, Kris Coombs, Shreya Havaldar, Gwenyth Portillo-Wightman, Elaine Gonzalez et al. 2018. TheGab Hate Corpus: A collection of 27k posts annotated for hate speech. *PsyArXiv* (2018). DOI: https://doi.org/doi:10.31234/osf.io/hqjxn

[49] Brendan Kennedy, Xisen Jin, Aida Mostafazadeh Davani, Morteza Dehghani, and Xiang Ren. 2020. Contextualizing hate speech classifiers with post-hoc explanation. *arXiv preprint arXiv:2005.02439* (2020).

[50] Douwe Kiela, Hamed Firooz, Aravind Mohan, Vedanuj Goswami, Amanpreet Singh, Pratik Ringshia, and Davide Testuggine. 2020. The hateful memes challenge: Detecting hate speech in multimodal memes. *arXiv preprint arXiv:2005.04790* (2020).

[51] Jae Yeon Kim, Carlos Ortiz, Sarah Nam, Sarah Santiago, and Vivek Datta. 2020. Intersectional Bias in Hate Speech and Abusive Language Datasets. arXiv:2005.05921 [cs.CL].

[52] Senthil Kumar, Aravindan Chandrabose, and Bharathi Raja Chakravarthi. 2021. An overview of fairness in data—Illuminating the bias in data pipeline. In *Proceedings of the 1st Workshop on Language Technology for Equality, Diversity*

and Inclusion. Association for Computational Linguistics, 34–45. Retrieved from https://aclanthology.org/2021.ltedi-1.5.

[53] Sachin Kumar, Shuly Wintner, Noah A. Smith, and Yulia Tsvetkov. 2019. Topics to avoid: Demoting latent confounds in text classification. *arXiv preprint arXiv:1909.00453* (2019).

[54] Ronan Le Bras, Swabha Swayamdipta, Chandra Bhagavatula, Rowan Zellers, Matthew Peters, Ashish Sabharwal, and Yejin Choi. 2020. Adversarial filters of dataset biases. In *Proceedings of the International Conference on Machine Learning*. PMLR, 1078–1088.

[55] Paul Pu Liang, Irene Mengze Li, Emily Zheng, Yao Chong Lim, Ruslan Salakhutdinov, and Louis-Philippe Morency. 2020. Towards debiasing sentence representations. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, 5502–5515. DOI: https://doi.org/10.18653/v1/2020.acl-main.488

[56] Haochen Liu, Wei Jin, Hamid Karimi, Zitao Liu, and Jiliang Tang. 2021. The authors matter: Understanding and mitigating implicit bias in deep text classification. In *Proceedings of the Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*. Association for Computational Linguistics, 74–85. DOI: https://doi.org/10.18653/v1/2021.findings-acl.7

[57] Florian Ludwig, Klara Dolos, Torsten Zesch, and Eleanor Hobley. 2022. Improving generalization of hate speech detection systems to novel target groups via domain adaptation. In *Proceedings of the 6th Workshop on Online Abuse and Harms (WOAH'22)*. Association for Computational Linguistics, 29–39. DOI: https://doi.org/10.18653/v1/2022.woah-1.4

[58] Raghvendra Mall, Mridul Nagpal, Joni Salminen, Hind Almerekhi, Soon-Gyo Jung, and Bernard J. Jansen. 2020. Four types of toxic people: Characterizing online users' toxicity over time. In *Proceedings of the 11th Nordic Conference on Human-Computer Interaction: Shaping Experiences, Shaping Society (NordiCHI'20)*. Association for Computing Machinery, New York, NY. DOI: https://doi.org/10.1145/3419249.3420142

[59] Binny Mathew, Ritam Dutt, Pawan Goyal, and Animesh Mukherjee. 2019. Spread of hate speech in online social media. In *Proceedings of the 10th ACM Conference on Web Science*. 173–182.

[60] Chandler May, Alex Wang, Shikha Bordia, Samuel R. Bowman, and Rachel Rudinger. 2019. On measuring social biases in sentence encoders. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, 622–628. DOI: https://doi.org/10.18653/v1/N19-1063

[61] Tomás Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. In *Proceedings of the 1st International Conference on Learning Representations*. Retrieved from: http://arxiv.org/abs/1301.3781.

[62] George A. Miller. 1995. WordNet: A lexical database for English. *Commun. ACM* 38, 11 (1995), 39–41.

[63] Pushkar Mishra, Marco Del Tredici, Helen Yannakoudakis, and Ekaterina Shutova. 2018. Author profiling for abuse detection. In *Proceedings of the 27th International Conference on Computational Linguistics*. 1088–1098.

[64] Marzieh Mozafari, Reza Farahbakhsh, and Noël Crespi. 2020. Hate speech detection and racial bias mitigation in social media based on BERT model. *PloS One* 15, 8 (2020), e0237861.

[65] Luke Munn. 2020. Angry by design: Toxic communication and technical architectures. *Human. Soc. Sci. Commun.* 7, 1 (July 2020). DOI: https://doi.org/10.1057/s41599-020-00550-7

[66] Jose I. Navarro. 2013. The psychology of hatred. *Open Criminol. J.* 6, 1 (Apr. 2013), 10–17. DOI: https://doi.org/10.2174/1874917801306010010

[67] Dat Quoc Nguyen, Thanh Vu, and Anh Tuan Nguyen. 2020. BERTweet: A pre-trained language model for English tweets. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing: System Demonstrations*. Association for Computational Linguistics, 9–14. DOI: https://doi.org/10.18653/v1/2020.emnlp-demos.2

[68] Debora Nozza, Claudia Volpetti, and Elisabetta Fersini. 2019. Unintended bias in misogyny detection. In *Proceedings of the IEEE/WIC/ACM International Conference on Web Intelligence*. 149–155.

[69] Nedjma Ousidhoum, Yangqiu Song, and Dit-Yan Yeung. 2020. Comparative evaluation of label agnostic selection bias in multilingual hate speech datasets. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP'20)*. 2532–2542.

[70] Ji Ho Park, Jamin Shin, and Pascale Fung. 2018. Reducing gender bias in abusive language detection. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2799–2804. DOI: https://doi.org/10.18653/v1/D18-1302

[71] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. GloVe: Global vectors for word representation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP'14)*. 1532–1543.

[72] Shraman Pramanick, Dimitar Dimitrov, Rituparna Mukherjee, Shivam Sharma, Md. Shad Akhtar, Preslav Nakov, and Tanmoy Chakraborty. 2021. Detecting harmful memes and their targets. In *Proceedings of the Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*. Association for Computational Linguistics, 2783–2796. DOI: https://doi.org/10.18653/v1/2021.findings-acl.246

[73] Shraman Pramanick, Shivam Sharma, Dimitar Dimitrov, Md. Shad Akhtar, Preslav Nakov, and Tanmoy Chakraborty. 2021. MOMENTA: A multimodal framework for detecting harmful memes and their targets. In *Proceedings of the Findings of the Association for Computational Linguistics: EMNLP 2021*. Association for Computational Linguistics, 4439–4455. Retrieved from: https://aclanthology.org/2021.findings-emnlp.379.

[74] Daniel Preoţiuc-Pietro and Lyle Ungar. 2018. User-level race and ethnicity predictors from Twitter text. In *Proceedings of the 27th International Conference on Computational Linguistics*. 1534–1545.

[75] Jing Qian, Mai ElSherief, Elizabeth Belding, and William Yang Wang. 2018. Leveraging intra-user and inter-user representation learning for automated hate speech detection. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, 118–123. DOI: https://doi.org/10.18653/v1/N18-2019

[76] Jing Qian, Hong Wang, Mai ElSherief, and Xifeng Yan. 2021. Lifelong learning of hate speech classification on social media. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, 2304–2314. DOI: https://doi.org/10.18653/v1/2021.naacl-main.183

[77] Md Mustafizur Rahman, Dinesh Balakrishnan, Dhiraj Murthy, Mucahid Kutlu, and Matthew Lease. 2021. An information retrieval approach to building datasets for hate speech detection. In *Proceedings of the 35th Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*. Retrieved from: https://openreview.net/forum?id=jI_BbL-qjJN.

[78] Dante Razo and Sandra Kübler. 2020. Investigating sampling bias in abusive language detection. In *Proceedings of the 4th Workshop on Online Abuse and Harms*. 70–78.

[79] Margaret E. Roberts, Brandon M. Stewart, Dustin Tingley, Christopher Lucas, Jetson Leder-Luis, Shana Kushner Gadarian, Bethany Albertson, and David G. Rand. 2014. Structural topic models for open-ended survey responses. *Amer. J. Polit. Sci.* 58, 4 (2014), 1064–1082.

[80] Jonathan Rosa. 2019. *Looking Like a Language, Sounding Like a Race*. Oxford University Press.

[81] Björn Ross, Michael Rist, Guillermo Carbonell, Benjamin Cabrera, Nils Kurowsky, and Michael Wojatzki. 2017. Measuring the reliability of hate speech annotations: The case of the European refugee crisis. *arXiv preprint arXiv:1701.08118* (2017).

[82] Paul Rottger, Bertie Vidgen, Dirk Hovy, and Janet Pierrehumbert. 2022. Two contrasting data annotation paradigms for subjective NLP tasks. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, 175–190. DOI: https://doi.org/10.18653/v1/2022.naacl-main.13

[83] Koustuv Saha, Eshwar Chandrasekharan, and Munmun De Choudhury. 2019. Prevalence and psychological effects of hateful speech in online college communities. In *Proceedings of the 10th ACM Conference on Web Science (WebSci'19)*. Association for Computing Machinery, New York, NY, 255–264. DOI: https://doi.org/10.1145/3292522.3326032

[84] Joni Salminen, Sercan Sengün, Juan Corporan, Soon Gyo Jung, and Bernard J. Jansen. 2020. Topic-driven toxicity: Exploring the relationship between online toxicity and news topics. *PLoS One* 15, 2 (Feb. 2020), e0228723. DOI: https://doi.org/10.1371/journal.pone.0228723

[85] Maarten Sap, Dallas Card, Saadia Gabriel, Yejin Choi, and Noah A. Smith. 2019. The risk of racial bias in hate speech detection. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. 1668–1678.

[86] Maarten Sap, Gregory Park, Johannes Eichstaedt, Margaret Kern, David Stillwell, Michal Kosinski, Lyle Ungar, and H. Andrew Schwartz. 2014. Developing age and gender predictive lexica over social media. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP'14)*. 1146–1151.

[87] Sheikh Muhammad Sarwar and Vanessa Murdock. 2022. Unsupervised domain adaptation for hate speech detection using a data augmentation approach. *Proc. Int. AAAI Conf. We Soc. Media* 16, 1 (May 2022), 852–862. Retrieved from: https://ojs.aaai.org/index.php/ICWSM/article/view/19340.

[88] Ulrich Schimmack. 2021. Invalid claims about the validity of implicit association tests by prisoners of the implicit social-cognition paradigm. *Perspect. Psycholog. Sci.* 16, 2 (Mar. 2021), 435–442. DOI: https://doi.org/10.1177/1745691621991860

[89] Anna Schmidt and Michael Wiegand. 2017. A survey on hate speech detection using natural language processing. In *Proceedings of the 5th International Workshop on Natural Language Processing for Social Media*. 1–10.

[90] Tal Schuster, Darsh Shah, Yun Jie Serene Yeo, Daniel Roberto Filizzola Ortiz, Enrico Santus, and Regina Barzilay. 2019. Towards debiasing fact verification models. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP'19)*. Association for Computational Linguistics, 3419–3425. DOI: https://doi.org/10.18653/v1/D19-1341

[91] Deven Santosh Shah, H. Andrew Schwartz, and Dirk Hovy. 2020. Predictive biases in natural language processing models: A conceptual framework and overview. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, 5248–5264. DOI: https://doi.org/10.18653/v1/2020.acl-main.468

[92] Arthur K. Spears. 2021. African-American language use: Ideology and so-called obscenity. *African-American English*, Routledge, 249–276. https://doi.org/10.4324/9781003165330-9

[93] Luc Steels. 2016. Human language is a culturally evolving system. *Psychonom. Bull. Rev.* 24, 1 (July 2016), 190–193. DOI: https://doi.org/10.3758/s13423-016-1086-6

[94] Julia Maria Struß, Melanie Siegel, Josef Ruppenhofer, Michael Wiegand, and Manfred Klenner. 2019. Overview of GermEval task 2, 2019 shared task on the identification of offensive language. *Preliminary Proceedings of the 15th Conference on Natural Language Processing (KONVENS'19, October 9–11, 2019 at Friedrich-Alexander-Universität Erlangen-Nürnberg)*, German Society for Computational Linguistics & Language Technology und Friedrich-Alexander-Universität Erlangen-Nürnberg, München [u.a.], 352–363. https://nbn-resolving.org/urn:nbn:de:bsz:mh39-93197.

[95] J. Suler. 2004. The online disinhibition effect. *Cyberpsychol. Behav.: Impact Internet, Multim. Virt. Real. Behav. Societ.* 7, 3 (2004), 321–326.

[96] Tony Sun, Andrew Gaut, Shirlyn Tang, Yuxin Huang, Mai ElSherief, Jieyu Zhao, Diba Mirza, Elizabeth Belding, Kai-Wei Chang, and William Yang Wang. 2019. Mitigating gender bias in natural language processing: Literature review. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, 1630–1640. DOI: https://doi.org/10.18653/v1/P19-1159

[97] Tony Sun, Andrew Gaut, Shirlyn Tang, Yuxin Huang, Mai ElSherief, Jieyu Zhao, Diba Mirza, Elizabeth Belding, Kai-Wei Chang, and William Yang Wang. 2019. Mitigating gender bias in natural language processing: Literature review. *arXiv preprint arXiv:1906.08976* (2019).

[98] Harini Suresh and John V. Guttag. 2019. A framework for understanding unintended consequences of machine learning. *arXiv preprint arXiv:1901.10002* (2019).

[99] Swabha Swayamdipta, Roy Schwartz, Nicholas Lourie, Yizhong Wang, Hannaneh Hajishirzi, Noah A. Smith, and Yejin Choi. 2020. Dataset cartography: Mapping and diagnosing datasets with training dynamics. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP'20)*. Association for Computational Linguistics, 9275–9293. DOI: https://doi.org/10.18653/v1/2020.emnlp-main.746

[100] Shawn Tan, Yikang Shen, Chin-wei Huang, and Aaron Courville. 2019. Investigating biases in textual entailment datasets. *arXiv preprint arXiv:1906.09635* (2019).

[101] Rachael Tatman. 2017. Gender and dialect bias in YouTube's automatic captions. In *Proceedings of the 1st ACL Workshop on Ethics in Natural Language Processing*. Association for Computational Linguistics, 53–59. DOI: https://doi.org/10.18653/v1/W17-1606

[102] Daniel Trottier. 2020. Denunciation and doxing: towards a conceptual model of digital vigilantism. *Global Crime* 21, 3-4 (2020), 196–212. DOI: https://doi.org/10.1080/17440572.2019.1591952

[103] Ameya Vaidya, Feng Mai, and Yue Ning. 2020. Empirical analysis of multi-task learning for reducing identity bias in toxic comment detection. *Proc. Int. AAAI Conf. Web Soc. Media* 14, 1 (May 2020), 683–693. Retrieved from: https://ojs.aaai.org/index.php/ICWSM/article/view/7334.

[104] Oskar Van Der Wal, Jaap Jumelet, Katrin Schulz, and Willem Zuidema. 2022. The birth of bias: A case study on the evolution of gender bias in an English language model. In *Proceedings of the 4th Workshop on Gender Bias in Natural Language Processing (GeBNLP'22)*. Association for Computational Linguistics, 75–75. DOI: https://doi.org/10.18653/v1/2022.gebnlp-1.8

[105] Juliet Van Rosendaal, Tommaso Caselli, and Malvina Nissim. 2020. Lower bias, higher density abusive language datasets: A recipe. In *Proceedings of the Workshop on Resources and Techniques for User and Author Profiling in Abusive Language*. 14–19.

[106] Sebastian Wachs, Manuel Gámez-Guadix, and Michelle F. Wright. 2022. Online hate speech victimization and depressive symptoms among adolescents: The protective role of resilience. *Cyberpsychol., Behav., Soc.l Netw.* 25, 7 (2022), 416–423. DOI: https://doi.org/10.1089/cyber.2022.0009

[107] Zeyu Wang, Klint Qinami, Ioannis Christos Karakozis, Kyle Genova, Prem Nair, Kenji Hata, and Olga Russakovsky. 2020. Towards fairness in visual recognition: Effective strategies for bias mitigation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 8919–8928.

[108] Zeerak Waseem. 2016. Are you a racist or am I seeing things? Annotator influence on hate speech detection on Twitter. In *Proceedings of the 1st Workshop on NLP and Computational Social Science*. 138–142.

[109] Zeerak Waseem and Dirk Hovy. 2016. Hateful symbols or hateful people? Predictive features for hate speech detection on Twitter. In *Proceedings of the NAACL Student Research Workshop*. 88–93.

[110] Laura Weidinger, John Mellor, Maribeth Rauh, Conor Griffin, Jonathan Uesato, Po-Sen Huang, Myra Cheng, Mia Glaese, Borja Balle, Atoosa Kasirzadeh, Zac Kenton, Sasha Brown, Will Hawkins, Tom Stepleton, Courtney Biles, Abeba Birhane, Julia Haas, Laura Rimell, Lisa Anne Hendricks, William Isaac, Sean Legassick, Geoffrey Irving, and Iason Gabriel. 2021. Ethical and Social Risks of Harm from Language Models. arXiv:2112.04359 [cs.CL].

[111] Maximilian Wich, Hala Al Kuwatly, and Georg Groh. 2020. Investigating annotator bias with a graph-based approach. In *Proceedings of the 4th Workshop on Online Abuse and Harms*. 191–199.

[112] Maximilian Wich, Jan Bauer, and Georg Groh. 2020. Impact of politically biased data on hate speech classification. In *Proceedings of the 4th Workshop on Online Abuse and Harms*. 54–64.

[113] Michael Wiegand, Josef Ruppenhofer, and Thomas Kleinbauer. 2019. Detection of abusive language: The problem of biased datasets. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. 602–608.

[114] Michael Wiegand, Melanie Siegel, and Josef Ruppenhofer. 2018. Overview of the GermEval 2018 shared task on the identification of offensive language, Austrian Academy of Sciences.

[115] Ellery Wulczyn, Nithum Thain, and Lucas Dixon. 2017. Ex machina: Personal attacks seen at scale. In *Proceedings of the 26th International Conference on World Wide Web*. 1391–1399.

[116] Mengzhou Xia, Anjalie Field, and Yulia Tsvetkov. 2020. Demoting racial bias in hate speech detection. In *Proceedings of the 8th International Workshop on Natural Language Processing for Social Media*. Association for Computational Linguistics, 7–14. DOI: https://doi.org/10.18653/v1/2020.socialnlp-1.2

[117] Wenjie Yin and Arkaitz Zubiaga. 2021. Towards generalisable hate speech detection: A review on obstacles and solutions. *PeerJ Comput. Sci.* 7 (2021), e598.

[118] Guanhua Zhang, Bing Bai, Junqi Zhang, Kun Bai, Conghui Zhu, and Tiejun Zhao. 2020. Demographics should not be the reason of toxicity: Mitigating discrimination in text classifications with instance weighting. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, 4134–4145. DOI: https://doi.org/10.18653/v1/2020.acl-main.380

[119] Jieyu Zhao, Tianlu Wang, Mark Yatskar, Vicente Ordonez, and Kai-Wei Chang. 2018. Gender bias in coreference resolution: Evaluation and debiasing methods. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, 15–20. DOI: https://doi.org/10.18653/v1/N18-2003

[120] Xuhui Zhou, Maarten Sap, Swabha Swayamdipta, Yejin Choi, and Noah Smith. 2021. Challenges in automated debiasing for toxic language detection. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*. Association for Computational Linguistics, 3143–3155. Retrieved from: https://www.aclweb.org/anthology/2021.eacl-main.274.

Received 26 January 2022; revised 26 December 2022; accepted 9 January 2023

Social Network Analysis and Mining (2022) 12:129
https://doi.org/10.1007/s13278-022-00951-3

REVIEW PAPER

Detection and moderation of detrimental content on social media platforms: current status and future directions

Vaishali U. Gongane[1,2] · **Mousami V. Munot**[1] · **Alwin D. Anuse**[3]

Received: 25 January 2022 / Revised: 6 August 2022 / Accepted: 8 August 2022 / Published online: 5 September 2022

Abstract

Social Media has become a vital component of every individual's life in society opening a preferred spectrum of virtual communication which provides an individual with a freedom to express their views and thoughts. While virtual communication through social media platforms is highly desirable and has become an inevitable component, the dark side of social media is observed in form of detrimental/objectionable content. The reported detrimental contents are fake news, rumors, hate speech, aggressive, and cyberbullying which raise up as a major concern in the society. Such detrimental content is affecting person's mental health and also resulted in loss which cannot be always recovered. So, detecting and moderating such content is a prime need of time. All social media platforms including Facebook, Twitter, and YouTube have made huge investments and also framed policies to detect and moderate such detrimental content. It is of paramount importance in the first place to detect such content. After successful detection, it should be moderated. With an overflowing increase in detrimental content on social media platforms, the current manual method to identify such content will never be enough. Manual and semi-automated moderation methods have reported limited success. A fully automated detection and moderation is a need of time to come up with the alarming detrimental content on social media. Artificial Intelligence (AI) has reached across all sectors and provided solutions to almost all problems, social media content detection and moderation is not an exception. So, AI-based methods like Natural Language Processing (NLP) with Machine Learning (ML) algorithms and Deep Neural Networks is rigorously deployed for detection and moderation of detrimental content on social media platforms. While detection of such content has been receiving good attention in the research community, moderation has received less attention. This research study spans into three parts wherein the first part emphasizes on the methods to detect the detrimental components using NLP. The second section describes about methods to moderate such content. The third part summarizes all observations to provide identified research gaps, unreported problems and provide research directions.

Keywords Social media (SM) platforms · Detection and moderation · Natural language processing (NLP) · Artificial intelligence (AI)

1 Introduction

In recent years, internet has revolutionized the communication domain through social media networks where people from different communities, culture and organization across the globe interact virtually. Internet has brought a dramatic change from web-based search engines to social media websites and micro-blogging sites which is gaining more popularity. Social media are defined as "a group of Internet-based applications that build on the ideological and technological foundations of Web 2.0, and that allow the creation and exchange of User-Generated Content" (Kaplan and Haenlein 2010). User Generated Content (UGC) describes the various forms of media content like text, video, and audio

✉ Vaishali U. Gongane
vug_entc@pvgcoet.ac.in

Mousami V. Munot
mvmunot@pict.edu

Alwin D. Anuse
alwin.anuse@mitwpu.edu.in

[1] E&TC Department, SCTR's Pune Institute of Computer Technology, SPPU, Pune 411043, India

[2] E&TC Department, Vidyarthi Griha's College of Engineering and Technology & G K Pate (Wani) Institute of Management, SPPU, Pune 411009, India

[3] School of ECE, Dr Vishwanath Karad MIT-WPU, Kothrud, Pune 411038, India

Table 1 Popular SM platforms

Name of SM platform	Category of SM	Year of launch	Characteristics of the platform
LinkedIn	Social networking site	2003	Professional networking website connect business people
			Used for used for professional networking and career development, and provide job opportunities
Facebook	Social networking site	2003	Enable users to stay connected with friends and relatives
			Users can chat, upload pictures, tell stories, share videos and links, post and read status
YouTube	Media sharing site	2005	Registered users can upload their content and share it with friends or provide it to the public
			Other users can rate and comment on the content
Twitter	Microblog	2006	Allows registered users to read and broadcast short messages called as tweets
			A tweet can be a text (140 characters), a photo or video content
WhatsApp	Messaging app	2009	Allows user to send text and voice messages, share images and videos
Instagram	Social networking site	2010	Designed to share photos and videos
			Users can create and share short videos with captions
Telegram	Messaging app	2013	Cloud based instant messaging and video calling service

created by the end users with commercial marketing context in mind. The UGC is published with either on publicly accessible website or social networking site accessible to certain group of people (Kaplan and Haenlein 2010). There are three aspects involved in definition of social media: Individuals create a public profile or a private profile. Second, individuals connect with friends, colleagues or relatives to form a network. Last, these individuals share their content and activities publicly in their network (Ellison 2007). All the three aspects are covered in various social networking sites like Facebook, Instagram, WhatsApp.

1.1 Various social media (SM) platforms

Before the invention of internet, SM began in the year 1844 with a series of electronic dots on a telegraph machine.[1] Bulletin Board Systems (BSS) was the first forms of SM that allowed users to log on and connect with each other. Usenet (USErNETwork) started by Tom Truscott and Jim Ellis in 1979 was a kind of discussion group where people can share views on topic of their interest and the article was available to all users in the group[1]. Six Degrees is considered to be the first social networking site similar to Facebook which had millions of registered users[1].

LiveJournal was a Weblog or blog publishing site that became popular in 1991. SM had various categories like blogs, forums, media sharing sites and social networking sites (Kaplan and Haenlein 2010). Table 1 shows the popular SM platforms[1] that have become an integral part of an individual's life. As shown in Table 1, the categories of SM provide the users to share the content in various formats.

Figure 1 shows the statistics of monthly active users on SM platforms up to year 2022. Facebook is the most widely used platform. In the first quarter of year 2022, Facebook had roughly 2.93 billion monthly active users.[2] SM can also serve as apparatus that assists many external and internal organizational activities among peer groups, customers, business partners, and organizations which include knowledge sharing, marketing strategies, product management, collaborative learning and sharing (Ngai et al. 2015).

Statistics report 43% of users search for products online through SM networks[3] indicating a new platform for private organizations to promote their brand and reach out to

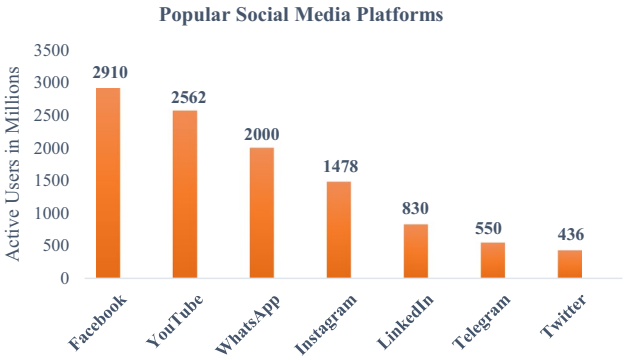


Fig.1 Statistics of monthly active users on various social media platforms (https://www.statista.com/statistics/278414/number-of-worldwide-social-network-users/)

[1] https://online.maryville.edu/blog/evolution-social-media/.

[2] https://www.statista.com/statistics/264810/number-of-monthly-active-facebook-users-worldwide.

[3] https://www.statista.com/statistics/278414/number-of-worldwide-social-network-users/.

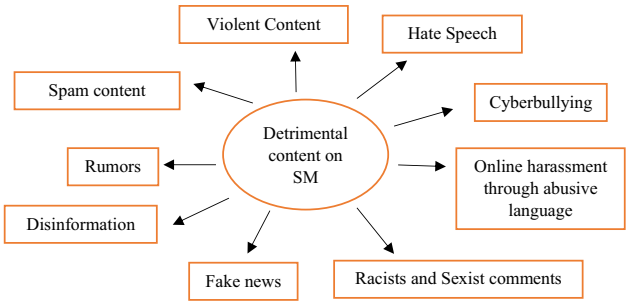


Fig. 2 Various forms of Detriment content published on SM

customers across the globe. For example, LinkedIn provides a platform for business to business and industry connectivity, career development activities, and job opportunities. There are also anonymous social networking mobile applications like Whisper where users post text messages and videos without revealing their identity.[4] The online social networking sites provide a platform for users to share their opinions on different aspects of social, political, economic, ethical, environmental issues in real time. The content called as User Generated Content (UGC) (Wyrwoll 2014) is shared on these platforms in form of text messages, images, videos, memes, audio. The terms like posts, tweets, comments, reviews, retweets are associated with UGC (Wyrwoll 2014). The content generated by the user is at times positive and at times is detrimental. The content on SM platforms is gaining importance through its use for screening students to provide placement opportunities and also used in a negative way that is affecting a person's mental health and also resulted in loss to economy. Recent years has shown an influential rise in the UGC on SM platforms which is creating a profound impact on the society.

1.2 The dark side of social media

Social media platforms like Twitter, Facebook, Reddit, and Instagram are the popular and widely used platforms that enable people to access and connect to a boundless world by forming a social network to express share and publish information (Nagi et al. 2015). Recent years have shown a substantial increase in the usage of SM platforms due to fast, easy access to information and a freedom to express through various formats (Wyrwoll 2014, Ruckenstein and Turunen 2020). This freedom of expression (Leerssen et al. 2020) is used in an improper way through the creation and publication of UGC that is provocative, inflammatory and threatening. In recent years the world is experiencing the negative aspect of SM through sharing of a detrimental content that is increasing at huge rate. A detrimental content on SM refers to sharing and publishing content with an intention to harm or distress a person or a community. Figure 2 depicts the detrimental form of UGC which includes hate speech content (Ayo et al. 2020), fake news, rumors (Shu et al. 2017), cyberbullying (Ofcom 2019), toxic content and child abuse material (Ofcom 2019) shown in Fig. 2. The definitions of the various forms of the detrimental/harmful content with an example content published on SM are depicted in Table 2. The term "fake news" on SM became prominent during US presidential election 2016. During the election period one of contenders made a speech: "The epidemic of malicious fake news and false propaganda that flooded social media over the past year. It's now clear that so-called fake news can have real-world consequences (Wendling 2018). As shown in Table 2, fake news, clickbait, rumors, satire news all come under misinformation (Islam et al. 2020) defined in context of two characteristics (Shu et al. 2017, Zhou et al. 2020):

(i) **Authenticity**: news that are non-factual or false which needs to be verified.
(ii) **Intent**: fake news is created with a wrong intention to mislead the users.

The authenticity characteristic cover disinformation, rumors, satire news and misinformation terms of fake news while the intent characteristic cover only disinformation and rumors. The current COVID-19 pandemic resulted in two million messages posted on Twitter with 7% of the total messages spreading conspiracy theories about the corona virus between 10 January 2020 and 20 February 2020 (Colomina et al. 2021).

Research studies have reported various definitions of hate speech like hate speech targets on a specific groups like ethnic origin, religion, or other, hate speech incite violence or hate toward minority, offensive and humorous content (Fortuna and Nunes 2018; Schmidt and Wiegand 2017). As shown in Table 2, hate speech content covers a broad spectrum of user created insulting words which are explored in various research works (Schmidt and Wiegand 2017). In many research articles, offensive content is also termed as abusive. Research articles have also reported the use of profane words in cyberbullying and hate speech content (Malmasi and Zampieri 2018). According to Pew Research survey 2018 conducted for teenagers, one in six teenagers have experienced one of the following forms of online abusive behavior as shown in Fig. 3.

The potential risk of SM has impacted the mental health of young generation in form of addiction, attention deficiency, aggressive behavior, depression, suicides (Ngai et al. 2015). According to National Crime Records Bureau (NCRB) data, cybercrimes are also increased on SM. In India, there are 578 cases of fake news on SM, 972 related

[4] https://en.wikipedia.org/wiki/Whisper_(app).

Table 2 Definition of various forms of inappropriate content published on SM

Name of the term	Definition	Example
Fake News	A news article that is intentionally false (Shu et al. 2017)	"*You see suicide rates are skyrocketing now*" (Patwa et al. 2021)
Deceptive News/ Disinformation	Deceptive news is articles with no correct facts, but articles are shared with authenticity (Shu et al. 2017, Zhou et al. 2020)	*Drinking hot water, cow urine, methanol or alcohol has been recommended as a proven cure for COVID-19* (Naeem et al. 2021)
Satire news	Satire news is form of fake news to attract the users with content written in a humorous and exaggerated way (Shu et al. 2017, Zhou et al. 2020)	*New UNL president a giant sea man* (Li et al. 2020)
Rumor	A piece of information that is shared on social media without being verified (Shu et al. 2017, Zhou et al. 2020)	*Saudi Arabia beheads first female robot citizen* (Islam et al. 2020, Ma et al. 2019)
Clickbait	Clickbait refers to attention grabbing form of news headlines on the social media (Shu et al. 2017, Zhou et al. 2020)	*This Rugby Fan's Super-Excited Reaction To Meeting Shane WilliamsWill Make You Grin Like A Fool* (Chakraborty et al. 2016)
Hate Speech:	The code of conduct as stated by European Union Commission: All public incitement to violence or hatred directed at a group of people or a member of that group based on race, color, religion, descent, nationality, or ethnicity (Fortuna and Nunes 2018)	*Refugees should face the figuring squad!* (Fortuna and Nunes 2018)
Cyberbullying	A form of harassment through electronic medium like mobile phone, computers conducted with an intention by a group or an individual against a person by sharing humiliating, wrong messages about him. This is a more general form of hate speech. (Fortuna and Nunes 2018)	*As long as fags don't bother me let them do what they want* (Dinakar et al. 2011)
Profanity	A sentence or a text with consists of offensive words or phrases. (Schmidt and Wiegand 2017)	*Holy shit, look at these ****** prices... damn!* (Malmasi and Zampieri 2018)
Toxic language	The toxic language is used in form of comments which include rude, disrespectful or unreasonable messages that can make other users to exit a discussion (Fortuna and Nunes 2018)	[a]*Ask Sityush to clean up his behavior than issue me nonsensical warnings*
Abusive language	A hurtful form of language that uses insulting or accusing words to someone but not targeted to a particular race, religion or ethnicity (Nobata et al. 2016)	*Add another JEW fined a billion for stealing like a lilmaggot. Hangthm all* (Nobata et al. 2016)
Sarcasm	Sarcasm is a form of ironic speech targeted to a particular victim to criticize him in a humorous way (Nobata et al. 2016)	*Most of them come north and are good at just mowing lawns* (Dinakar et al. 2011)

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

to cyber bullying of women and children, and 149 incidents of fake profile as reported in Times of India 2020. The recent example of COVID-19 pandemic has resulted in 80% of users reading fake news about the outbreak of the corona virus.[5] There were 7 million fake news stories, 9 million content encouraging extremist organization and 23 million hate speech content that were removed by SM companies during COVID-19 pandemic. This forced the European Commission (EU) to frame the policies to tackle the growing online threats and misinformation. The World Health Organization (WHO) reported that the citizens around the globe were victims of pandemic and "infodemic" that came up along with it 2020 (Colomina et al. 2021; Nascimento

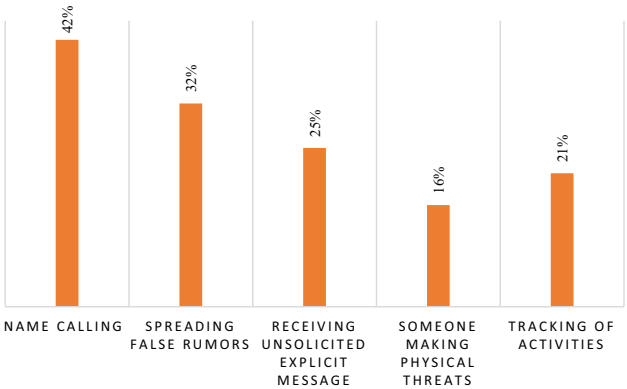


Fig. 3 Online abusive behavior experienced by teenagers

[5] https://www.thenews.com/ Interesting statistics about fake news on social media/print/893091.

Table 3 Legal provisions to tackle the misuse of SM

Name of legal provision	Points covered in the provision	Remarks
The Information Technology (Amendment) Act, 2008 Sect. 66A (Mangalam and Kumar 2019)	Prohibits sending of offensive messages through an online medium	Declared 'unconstitutional' in 2015 due to lack of interpretation of terms like 'grossly offensive', 'insult', menacing' in the Sect. 66A
	Person sending information for the purpose of causing annoyance, obstruction, hatred or ill will through any digital platform is punishable with imprisonment for more than 3 years and with fine	Supreme Court found it violative of right to freedom of expression under Article 19 (1)
The Information Technology (Guidelines for Intermediaries and Digital Media Ethics Code) Rules, 2021[a]	Guidelines for SM intermediaries to identify the 'originator' of 'unlawful' messages	Differentiates between social media intermediary and significant social media intermediary
	Intermediaries shall remove or disable access within 24 h of receipt of complaints of sexual act morphed images etc	Challenged by SM companies as limit to freedom of expression

[a]G.S.R. 139(E): the Information Technology (Intermediary Guidelines and Digital Media Ethics Code) Rules, 2021.https://www.meity.gov.in/content/notification-dated-25th-february-2021-gsr-139e-information-technology- intermediary

et al., 2022). The definition of hate speech is subjective and varies with context in which the words are used in the content and is highly dependent on the geographic location.

1.3 Legal Provisions made by Government and SM companies to tackle the detrimental content

To curb with the increasing detrimental content on SM, Government has made legal provisions for example IT ACT 2000 law in India to deal with cybercrime and electronic commerce. The legal provisions defined by Government of India are summarized in Table 3.

As shown in Table 3, the compliance with the points defined in the legal provisions is challenging in view to safeguard the right of freedom of speech and expression of an individual on SM and a need to define what form of content is offensive or insulting. The Guidelines for Intermediaries and Digital Media Ethics Code) Rules show stringent guidelines for intermediaries in terms of taking down the 'unlawful' messages within a specific timeframe and providing information related to originator of such message and verification of identity to authorized agencies within 72 h.[6] This aspect though may help to control the spread of such messages but will be taken into account after such'unlawful' messages are flooded on SM and a damage is caused in the society. Government has also framed the legal rules and policies for SM companies that need to be implemented when an objectionable post published on online platforms. The rules are also defined for SM companies when an objectionable posts results in disturbing incidents and cause damage. The SM companies take counter actions by either removing or deleting the posts or by blocking the account of

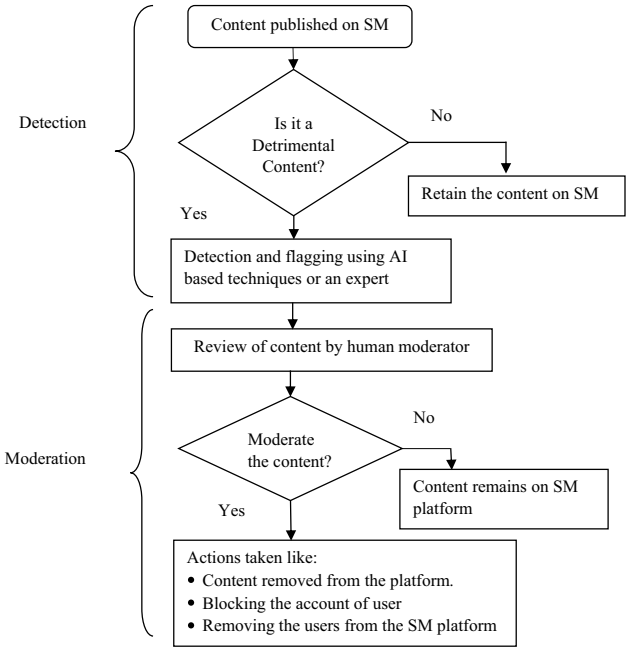


Fig. 4 Detection and moderation of UGC on SM platforms

the user who published the posts (Roberts 2017b). For example, Twitter platform received 1698 complaints pertaining to online abuse/harassment (1366), hateful conduct (111), misinformation and manipulated media (36), sensitive adult content (28), impersonation (25) in India via its local grievance mechanism between April 26, 2022 and May 25, 2022.[7] The action taken by Twitter is either in form of removing the accounts or banning the accounts that promote such activities.

[6] G.S.R. 139(E): the Information Technology (Intermediary Guidelines and Digital Media Ethics Code) Rules, 2021.https://www.meity.gov.in/content/notification-dated-25th-february-2021-gsr-139e-information-technology- intermediary.

[7] https://www.business-standard.com/article/companies/twitter-says-it-has-banned-over-46-000-bad-accounts-in-india-in-may-122070300540_1.html.

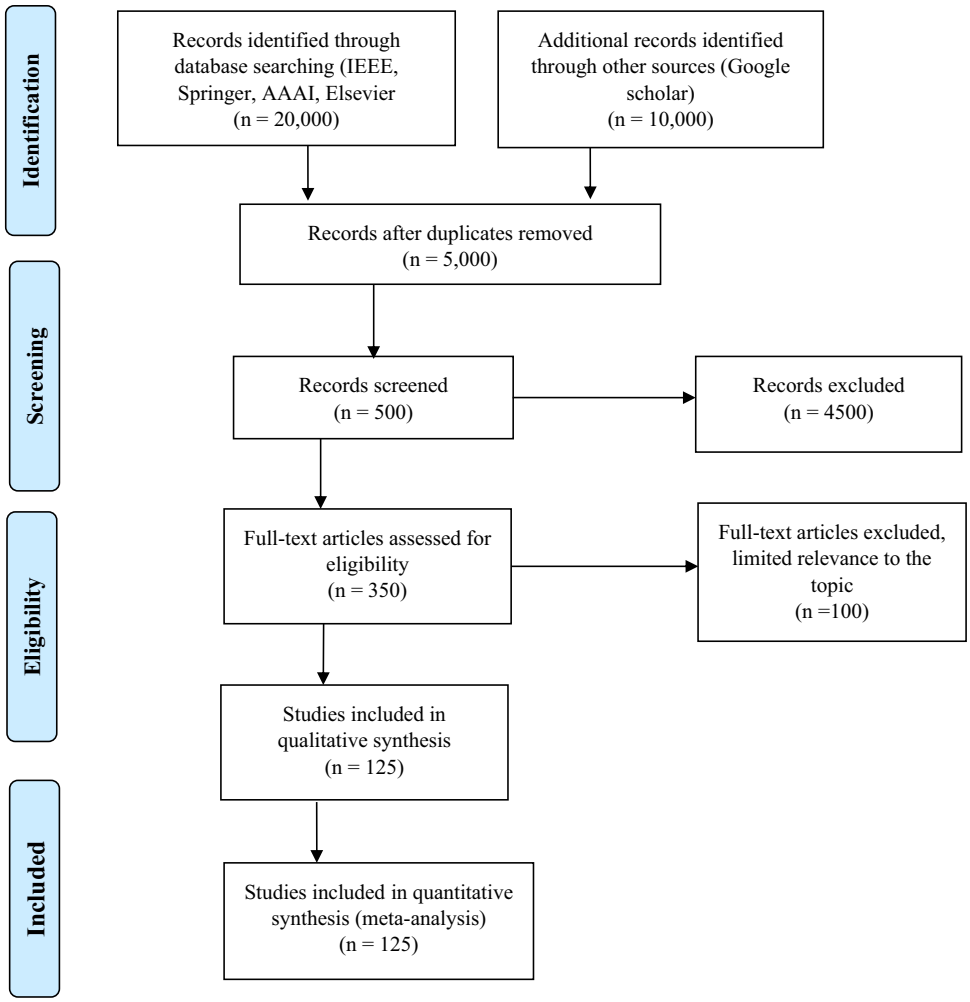


Fig. 5 Flow chart of selection of articles for review

1.4 Detection and moderation of detrimental content on SM

Considering the huge volume of UGC on various SM platforms, detection and moderation of detrimental content on SM is of paramount importance. When a content published on SM platforms, it is detected to identify or classify whether the published content is harmful or non-harmful. Figure 4 depicts the steps of UGC detection and moderation on SM platforms. Detection is a task of classifying UGC as a normal content or an inappropriate content. Detection method entails identifying: the slur or slang, abusive, profane words in the content and the fake news in the content, and checking whether the content is targeting to a particular community or an individual. Artificial Intelligence (AI) has emerged as an upcoming tool for automated detection of detrimental content on SM through Machine Learning (ML) algorithms and Natural Language Processing (NLP). The use of AI-based detection methods assists the human moderators in flagging the content. UGC moderation on SM platform is the systematic screening of User Generated Content (UGC) provided to websites, SM, and other online networks to determine the content's acceptability for a specific site, location, or jurisdiction (Roberts 2017a). Moderation is about making a decision about the checking and verifying the adequacy of the detected content according to the rules and policies as defined by a particular SM platform. So, moderation is with respect to a specific SM platform. For example, a dance video published on LinkedIn is unacceptable as it is a professional SM platform with emphasis on building a network of professionals from various industries across the globe. The same dance video is acceptable on Facebook as it promotes sharing of individual user content in various forms. So, content moderation is more dependent on SM platform.

1.5 Organization of the paper

The paper is organized as follows: Sect. 2 describes the Review methodology used for presenting the paper. Section 3 presents the datasets created by research community for UGC detection on SM platforms. Section 4 provides the

UGC detection and next section presents UGC moderation. The article concludes with conclusion and directions for further research.

2 Review methodology

A systematic method of reviewing the available literature is adopted to explore the work done by researchers in the field of SM content moderation. The methodology of literature review is divided into following steps:

- Defining the research questions
- Collection of relevant topics from the scientific literature and recent articles.
- Mapping the information collected from the literature to the research questions.

Figure 5 shows the flow diagram of the selection process of research articles for the review. With an objective of understanding the SM content detection and moderation, an ordered process of search is utilized with research articles collected from various fields of social sciences, computational intelligence and technology. The literature survey for the study was restricted to the articles published during the year 2011–2021. With reference to the objective of the study, the first step consists of collecting the articles from IEEE, Springer, Elsevier and AAAI digital libraries and Google Scholar. Since Google Scholar consists of articles from all publishers, including Arxiv, duplicate articles were excluded. A total of 500 articles related to social media content were screened by reading the abstract of the article and the maximum number of citations received for the article. The process of collecting the articles by exploring the literature in domain of social sciences with keywords like "Content moderation on social media", "User generated content on social media", and "Need of content moderation" in the digital library database. This research paper focuses on detection and moderation of detrimental content on social media, so after giving this query on Google scholar resulted in articles related to detection of hate speech, fake news, rumors and cyberbullying content. On this basis, queries like "Detection of harmful/problematic social media content using Natural Language Processing", "Machine learning and Deep Learning algorithms for Hate Speech/Fake news/rumors", "NLP for Hate Speech/Fake news/rumors detection", "Hate Speech/Fake news/rumors detection using machine learning and deep learning techniques" were investigated on digital libraries. For query related to social media content moderation, majority of the articles were extracted from social science domain. Considering detection and moderation of SM content, a total of 125 articles were selected in this study (Fig. 5).

2.1 Research objectives

The study presents an exhaustive survey of research done in SM content detection and moderation techniques. The key research objectives of the study are to:

- Outline the various forms of detrimental contents like rumors, fake news, hate speech, abusive content which exemplify the inappropriate use of SM.
- Review the datasets used for detection of detrimental content.
- Perform a comparative analysis of various Language Models (LM) and Machine Learning (ML) algorithms used for detection of detrimental content on SM platforms.
- Review the moderation techniques of detrimental content.
- Identify the challenges and research gaps of various reported techniques for UGC detection and moderation.

2.2 Research questions

The following research questions are framed to meet the research objectives.

Which datasets are used for detrimental content detection techniques?
What are the various methods to detect detrimental content on social media platforms?
What is content moderation and approaches to content moderation on social media platforms?
What are the challenges and research gaps in the reported techniques for content detection and moderation?

The paper is organized with the sections corresponding to meeting the defined objectives and answering the framed research questions.

2.3 Theoretical and practical implications of study (Cunha et al. 2021)

The literature review shows that there is a massive amount of research explored in the detection methods of various forms of detrimental content. From theoretical point of view, reported articles have focused more on the various aspects involved in terms of manual method of moderation and challenges that the AI based methods should address. There are less research articles that focus on fully automated moderation techniques of detrimental content on social media platforms. From practical point of view, more experimentations

Table 4 Popular datasets for fake news detection

Dataset	Features	Categories of labels assigned to articles	Skewness (Cunha et al. 2021)
LIAR (Wang 2017)	First dataset for deception detection	False (pants-fire)	Highly imbalanced
	12.8 K human labeled short statements evaluated PolitiFact.com	False	
		Barely true	
		Half true	
	Statement collected from news releases, TV/radio interviews, campaign speeches, TV ads, tweets, debates, Facebook posts, etc	Mostly true	
		True	
BUZZFEED NEWS[a]	2000 news samples published on Facebook during 2016 US Presidential elections	Mostly true	Highly imbalanced
		Not factual content,	
	Each post and linked articles were checked by 5 journalists	Mixture of true and false	
	Metadata information such as URL of the news post, published data, number of shares, reactions and comments	Mostly false	
CREDBANK (Mitra and Gilbert 2015)	60 million tweets from Twitter which covers 1049 real-world events	Real	Imbalanced
	Credibility verified by 30 annotators from Amazon Mechanical Turk	Fake	
FAKENEWSNET (Lee et al. 2018)	Highlights on dynamic context and social behavior of fake news	True	Balanced
	211 fake news and 211 true news	Fake	
	Data like publisher information, news content, and social engagements information gathered from fact checking websites BuzzFeed.com and PolitiFact.com		
Fake News Challenge-FNC-1[b]	Used for stance detection method with 50,000 stances, which targets on estimating the stance of a body text from a news article relative to a headline	Agrees	Highly imbalanced
		Disagrees	
		Discusses, or	
		Unrelated	
ISOT (Ahmad et al. 2017)	Real and fake news articles	Real-21417	Slightly imbalanced
	Real news gathered from crawling website Reuters.com	Fake-23481	
	Fake news from unreliable websites flagged by Politifact and Wikipedia		
	Include articles of political and World news topics		

[a]https://github.com/BuzzFeedNews/2016-10-facebookfact-check/tree/master/data

[b]https://github.com/FakeNewsChallenge/fnc-1

are done on language models, non-neural and neural network models for detection of detrimental content.

3 Datasets

Datasets form an important repository which contains information in form of a table. In context of detrimental form, the information in the datasets includes news articles, URLs, slang words, publisher information, social engagements, tweets gathered from social media platforms. Various ML algorithms are experimented on the available datasets for detection of fake news, hate speech and its related terms.

The datasets for fake news are prepared by extraction of online comments or posts from various social media platforms. The datasets are created with help of language experts and experts from field of journalism. The human experts analyze the posts and comments and assign labels to them as fake and real. Table 4 compares the list of features that can be extracted from the available datasets for fake news detection. As seen from Table 4, most of dataset's target on the content features of the news, which might be not sufficient for an effective detection of fake news. Datasets like BuzzFeed News and FNC-1 and Fake News Net include metadata information and also the news content

Table 5 Popular datasets for hate speech detection

Name of Dataset	Features	Categories of labels assigned to articles	Skewness (Cunha et al. 2021)
Davidson et al. (2017)	24,802 tweets from Hatebase	Hate speech-7%,	Highly imbalanced
	Contain large number of ethnicity content	Not offensive-,	
	Collection on offensive keywords	Offensive but not hate speech	
Stormfront (Bonet et al. 2018)	Textual hate speech annotated at sentence level	Hate,	Imbalanced
		No hate	
		Relation	
	10,568 sentences have been extracted from Stormfront	Skip	
ETHOS (Mollas et al. 2021)	Creation of two textual datasets using comments from Reddit and Youtube	Violence	Highly Imbalanced
		Directed_vs_generalized	
	First dataset includes 998 comments with two labels	Gender	
		Race	
	Second dataset includes 433 hate speech messages with 8 labels	National_origin	
		Disability	
		Sexual_orientation	
		Religion	
Hatebase[a]	Online repository of structured and usage-based hate speech	Archaic	Highly Imbalanced
		Class	
	Used to build a classifier for hate speech	Disability,	
		Ethnicity,	
		Gender,	
		Nationality,	
		Religion	
		Sexual orientation	
HASOC 2019 (Mandl et al. 2019)	Three datasets from Twitter and Facebook	Hate and offensive	Imbalanced
		Profane	
	German, English and Hindi language	Non- hate and offensive	
CONANN (Chung et al. 2019)	Expert based hate speech and counter narrative content	Hate speech	Imbalanced
	4078 pairs over the English, French and Italian language	Counter hate speech	
	Expert demographics, hate speech sub-topic and counter-narrative type		
Waseem and Hovy (2016)	136,052 tweets from Twitter Labeling of 16,192 tweets	Racist	Imbalanced
		Sexist	
		None	
Waseem and Hovy (2016b)	136,000 tweets from Twitter Annotations by experts (feminists and anti-racism activists) and crowdsource workers	Racist	Highly Imbalanced
		Sexist	
		None	
		Both	
TRAC (Ojha et al. 2018)	15,000 instances from Twitter and Facebook	Overtly aggressive	Highly imbalanced
		Covertly aggressive	
	4 different datasets for English and Hindi language	Non-aggressive	

Table 5 (continued)

Name of Dataset	Features	Categories of labels assigned to articles	Skewness (Cunha et al. 2021)
GERMEVAL (Ross et al. 2016)	5009 tweets from Twitter in German	Offensive	Imbalanced
	Shared task with binary classification and fine-grained classification	Profanity	
		Abuse	
		Other	
SemEval 2019 hatEval (Basile et al. 2019)	13,240 tweets from Twitter	Hateful	Imbalanced
	Hateful content against immigrants and women in English and Spanish	Aggressive	
KAGGLE[b]	8832 social media comments	Insulting	Imbalanced
		Non-insulting	
Golbeck et al. (2017)	20,362 tweets from Twitter	Positive Harassment	Imbalanced
		Negative Harassment	
HOET (Mathur et al. 2018)	3679 tweets in Hindi-English code-switched language	Not Offensive	Highly Imbalanced
		Abusive	
Hindi-english offensive tweet		Hate-Inducing	

[a] www.hatebase.org

[b] https://kaggle.com/c/detecting-insults-insocial-commentary

features which are explored in many research articles. The metadata information includes social network information, user's engagements in the news, users' profiles, etc. (Shu et al. 2017). LIAR dataset has considerably huge statements as compared to other datasets and also include meta-data information of each speaker (Wang 2017). The LIAR dataset also covers diverse subject topics like economy, healthcare, taxes, federal-budget, education, jobs, state budget, candidates-biography, elections, and immigration. Some datasets also assign labels to news articles to enable have a multi-level classification.

Table 5 summarizes the datasets for various forms of hate speech. The datasets of hate speech contain monolingual and multilingual content and also include score labels (Davidson et al. 2017) assigned to each characteristic of hate speech. The annotation of hate speech is done by different annotators; the evaluation of which is done by a metric called inter-annotator agreement (Nobata et al. 2016; Davidson et al. 2017; Kocoń et al. 2021). The inter annotator agreement defines the number of annotators that agree on a particular task of annotation (Kocoń et al. 2021). Fleiss's Kappa (κ) is a statistical metric that specifies the annotators rating for assigning a label to content (Davidson et al. 2017) and Krippendorff's alpha (Singhania et al. 2017) deals with annotations that are missed. These two measures are utilized for datasets with a high value of this measure signifies higher level of agreement. For example, Vigna et al 2017) reported a $\kappa = 0.26$ for 1687 comments annotated by 5 annotators for 2 classes of hate speech: weak hate and strong hate which shows the difficulty in annotation process. Nobata et al. (2016) reported a $\kappa = 0.26$ for 56,280 abusive comments annotated by 3 expert raters. Waseem et al. (2016) reported a $\kappa = 0.84$ with 85% disagreement for annotations of sexism. Due to highly subjective nature of hate speech, the inter-annotator agreement process becomes too challenging. Many research studies reported creation of datasets that assign labels as offensive, abusive, profanity, racism, sexism and general hate. As seen in Table 5, the skewness level (Cunha et al. 2021) of only few datasets is balanced. For example, the hatEval (Basile et al. 2019) dataset include 43% as hate content and 57% as non-hate content. Davidson et al. (2017) reported 5% as hate speech and 76% as offensive language. The label "relation" in (Bonet et al. 2018) indicates a hate speech sentence when it is combined with other sentences and label "skip" signifies a non-English sentence or a sentence with a hate or non-hate speech.

The inter-annotator agreement plays a vital role in creating the datasets for hate speech as it affects the performance of a ML algorithm (Kocoń et al. 2021). In context of fake news and hate speech, Twitter is the preferred social media platform for extracting information and preparing a dataset. The creation of datasets is dependent on the annotator's perspective of assigning a label and context information about the content. With a tendency of a user to write a post in multilingual form and code-mixed form (native language written in Roman script), research community have also created datasets in code-mixed language (Hindi + Engish) (Mathur et al. 2018) which are used for detection of hate speech using ML and neural network architectures. The annotation of such form of content is done with human annotators

Table 6 Fact-checking websites

Fact Checking website	Description
Politifact.com[a]	US-based website for fact checking of political news and information
	Provide labels to the claims and statements as True, Mostly true, Half true, Mostly false, False, and Pants on fire
	Experts verify the creditability of the statement and claims by examining a specific word and the context of the claim
Snopes.com[b]	First online fact checking website considered by many journalists and researchers for verifying internet rumors and misinformation
	Website covers various subjects like medicine, science, history, crime, frauds, etc
	Websites provide a comprehensive evaluation of various types of printed resources and assign truth ratings to them based on the knowledge from professional individuals and organizations
FactCheck.org[c]	Website evaluates the truthfulness of the facts and claims made during the election years by U.S. political players on various platforms like television, social media, speeches and interviews
	With help of experts, each piece of information is checked and analyzed in a systematic manner
Hoax-slayer.net[d]	Website has debunked email and internet hoaxes, thwarted scammers, educated web users about security issues and combated spam based on meticulous research on the information gathered from news articles, press release, government publications, reputed websites
	Websites include articles that include hyperlinks and reference list that allow the reader to check the information for themselves

[a] www.politifact.com

[b] www.snopes.com

[c] https://www.factcheck.org

[d] www.hoax-slayer.net

and inter annotator agreement is calculated. As shown in Table 5, there are too diverse variations in the datasets of hate speech, for example: "aggressive" word with labels like covertly and overtly aggressive and "hate-inducing" word. The multiple labels assigned to the text in the datasets, the size of datasets and skewness level affect the performance of ML algorithms and deep neural network models. Research articles have reported few questionable and doubtful cases (Mathur et al. 2018) of hate speech that were too challenging for human annotators to decide. Such cases were not considered in the dataset. Such uncertain cases need to be addressed in the dataset.

4 Detection of detrimental UGC on SM

Detection is a task of identifying the detrimental or objectionable content from the posts or text messages published by users on SM platforms. Detecting the detrimental content includes identifying the fake news, hate speech, abusive language content in an online post. Before moderating the content on SM platforms, it is first detected. Considering the amount of content published on SM platforms, (for example: An average 6000 tweets are posted every second on Twitter[8]), manual method of detection is not scalable. Artificial intelligence (AI) has emerged as an important tool for identifying and filtering out UGC that is offensive or harmful. Various techniques of AI in form of ML algorithms, DL, and Natural Language Processing (NLP) are deployed for detection of detrimental UGC (Ofcom 2019; Grimmelmann 2015). Research articles have reported that the AI based tools have achieved optimal accuracy and speed in detecting the detrimental content on SM platforms. This section describes the manual and AI-based methods of detecting detrimental content on SM platforms.

4.1 Manual method of fake news detection

Fact checking is a detection method that decides of whether the published content is real or fake (Barrett 2020). Fact checking does not evaluate an objectionable content, but classifying whether the content is true or false (Barrett 2020).

Table 6 depicts the fact-checking websites. Fact checking websites make use of human experts in the journalism domain that check the veracity of the news content.

The experts are called as fact checkers that follow a methodology to evaluate a content. The methodology utilized by fact-checking websites includes:

(i) By skimming through news items, political commercials and speeches, campaign websites, social media, and press releases, TVs, and interviews, a topic or a claim to be examined is chosen.

[8] https://www.internetlivestats.com/twitter-statistics/.

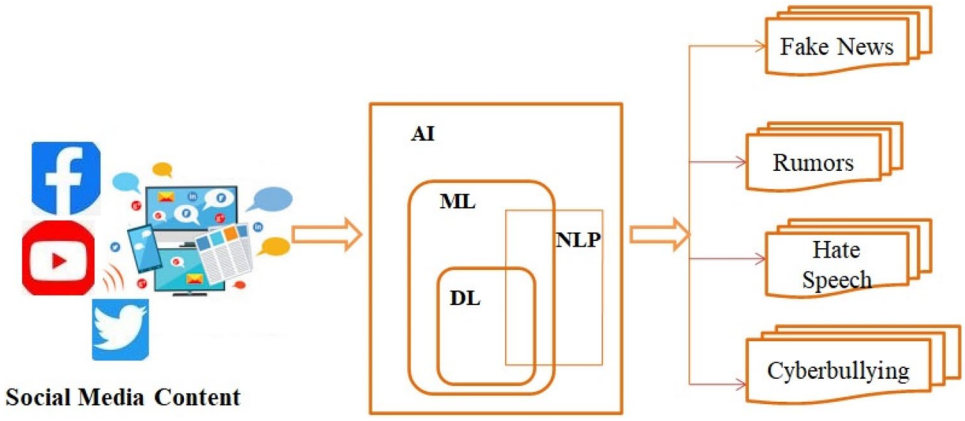


Fig. 6 AI-based techniques for detection of detrimental content on SM platforms

(ii) Fact checkers most typically employ fundamental methodologies and types of sources while conducting research on assertions, as well as official regulations and editorial norms that govern their approaches.
(iii) Claim assessments, which are systems and processes used by fact-checkers to determine the validity of a claim.[9]

The fact checking website like Politifact[9] has developed datasets and made it publicly available for automatic detection of fake news content. These websites provide an expert analysis for checked news as which news articles are fake and reason for why it is fake (Zhou and Zafarani 2020). SM platforms like Facebook sends flagged content to more than 60 fact-checking organizations worldwide, but each organization typically assigns only a handful of reporters to investigate Facebook posts (Barrett 2020). The manual method of checking the facts for detecting the fake news is a complex task. Factors like time needed to check the veracity of the news and the knowledge of the context around the fake news need to be considered in the detection task.

The detection of other forms of detrimental content like hate speech and abusive language is done by the user community to express their concern about the content posted on SM platforms (Gillespie 2018, Crawford and Gillespie 2016). There is risk of bias getting introduced by the user in the detection of such content. With overflowing increase in detrimental content, the manual method of detection will not be adequate.

4.2 Detection of detrimental UGC using natural language processing (NLP)

The manual approach of fake news detection has many challenges in terms of the volume, veracity and speed of content to be analyzed, the cultural, historical and geographical context around the content. Many companies and governments are proposing automated processes to assist in detection and analysis of problematic content, including disinformation, hate speech, and terrorist propaganda (Leerssen et al. 2020). Past decade has shown significant developments in AI through the advances in the algorithms, computational power and data (Ofcom 2019). Deep Learning (DL) is a subfield of ML that makes use of Artificial Neural Networks (ANN) to process huge amount of data. Natural Language Processing (NLP) is a subfield of AI that uses techniques to parse the text using computers (Hirschberg and Manning 2015). Natural Language Processing (NLP) is a computational linguistic field which makes use of computational techniques to learn and understand human language (Hirschberg and Manning 2015).

ML, ANN and NLP are the key components that have contributed to automated detection of detrimental form of SM content. Figure 6 shows the AI based approach of detection of detrimental content on SM platforms. A large volume of research has been explored on use of AI based techniques for detection of fake news, rumors, abusive/offensive language, and hate speech on SM platforms. The task of automated detection of UGC using NLP, ML and DL algorithms consists classifying the online comments/posts as detrimental (which include hate speech, abusive, toxic, rumors, cyberbullying) or a normal content. NLP has opened new spectrum of automating the linguistic structure of language in creation of speech-to-speech translation engines, mining SM for information about health or finance, and identifying sentiment and emotion toward products and services (Hirschberg and Manning 2015), filtering offensive content, and improving spam detection (Duarte et al. 2017), creation of chatbots for customer service (Ofcom 2019).

The noteworthy advancements in NLP have played a major role in detection of detrimental content on SM platforms. NLP tools are widely used to process the text-based online comments on SM (Ofcom 2019). In context of

[9] www.politifact.com.

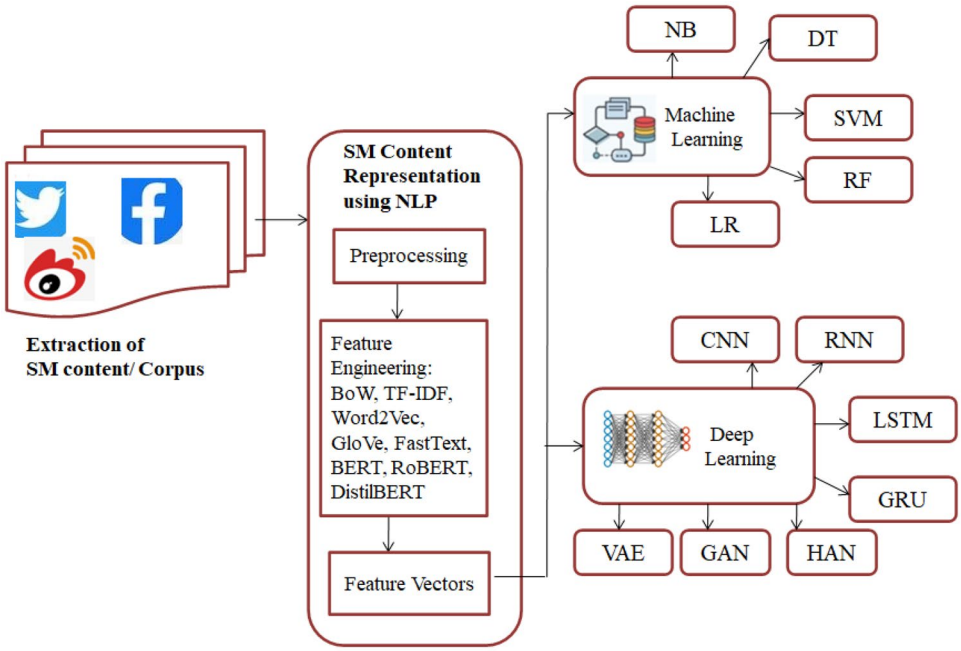


Fig. 7 A generic block diagram of automated SM content detection using NLP, ML and DL

content moderation, NLP techniques are used to process the online text, extract the features from text which are used to detect the harmful forms content like fake news, hate speech, cyberbullying.

Recent years have shown advancements in NLP tools working as text classifiers that use neural networks and ML to analyze the features of text and classify the text into one of the categories of detrimental content and normal content (Duarte et al. 2017). Considering the amount of SM content, analysis of content using NLP includes quantitative and qualitative analysis. Quantitative analysis makes use of statistical measures like counting the frequency of words in content. Qualitative analysis investigates the meaning and semantic relationship of words and phrases in the content. Figure 7 depicts a generalize block diagram of UGC detection. NLP tools are deployed to process the online content published on the SM platforms. As shown in Fig. 7, the extraction of SM content comprises of acquisition of online comments and posts through Application Programming Interface (API) and crawling methods provided by SM platforms. For example, Twitter provides two tools namely the Search and Streaming API to collect the data (Ayo et al. 2020). A corpus is created that covers all diverse forms of SM content in monolingual and multi-lingual configuration with metadata information like geographical location, user profiles and followers (Schmidt and Wiegand 2017; Duarte et al. 2017).

This corpus is created with help of experts and crowdsource workers that assign labels to content as a normal one or harmful one (Roberts 2017a). The corpus thus created is called as dataset and researchers have made a significant contribution in creation of dataset that covers all terminologies of detrimental content like fake news, rumors, hate speech and cyberbullying content. The comment features are extracted from the corpus using NLP tools. The features can be words, phrases, characters, unique words (Schmidt and Wiegand 2017; Ahmed et al. 2017) that differ depending on the form of content to be processed. Many feature representation techniques like Bag of Words (BoW), Term Frequency-Inverse Document Frequency, n-grams (Schmidt and Wiegand 2017; Ahmed et al. 2017), Word2Vec (Mikolov et al. 2013), GloVe (Pennington et al. 2015), Bidirectional Encoder Representation from Transformer (BERT) (Vaswani et al. 2017; Devlin et al. 2019) map the text features from the content to vectors of real numbers known as feature vectors.

The feature vectors obtained after processing the SM content using NLP tools are applied to a classifier model which can be either a non-neural model or a neural model (Cunha et al. 2021). Classifier models are used to detect detrimental content based on features extracted from SM content. Research literature reports the use of supervised ML algorithms like Support Vector Machine (SVM), Logistic Regression (LR), Naïve Bayes (NB), and Random Forest (RF) and deep neural networks like non-Sequential neural network models: Convolutional Neural Networks (CNN) and sequential neural network models: Recurrent Neural Networks (RNN), Long Short Term Memory (LSTM), Gated Recurrent Unit (GRU), Transformer models, Variational Autoencoder (VAE) models, and Graph based neural networks for the detection and classification of detrimental SM content which predominantly include fake news and hate speech. The non-neural and neural network models (Cunha

Table 7 Pre-processing Steps (Vijayarani et al. 2015, Ahmed et al. 2017, Robinson et al. 2018, Elhadad et al. 2020)

Pre-processing step	Description
Conversion of the text to lower case	All words in the text are written in lower case to eliminate the difference between the letters of the same words
	The words "GOOD" and "gOoD" convey the same meaning but written in mixed casing style
	Lower casing of the entire text makes analysis easier and leads to reduction in dimensionality of data
Removal of Stop Words	Stop words are articles ('a', 'an', 'the') and prepositions ('for', 'on') in the text and convey no useful meaning
	Stop words are not considered as keywords in text analysis so they are removed
Stemming	Rule-based method of reducing inflectional form of a word to its base or root form
	Common stemming algorithms include Porter's Stemming, Dawson Stemmer, N-gram Stemmer
	Removal of suffixes from a word which reduces the dimensions of data in terms of space and memory
	For example, words like playing, played are related to a word "play" with same meaning
Lemmatization	Lemmatization identifies the base form of the word through morphological analysis
	Lemmatization always provides a dictionary meaning of a word
Tokenization	Tokenization is a method of breaking the raw text into tokens or words and is language dependent
	A Python tokenization package named "Twikenizer" is capable of dealing with special characters attached to the words

et al. 2021) are trained on various features extracted from the labeled datasets using various feature representation techniques. The trained network is applied on the test data for detection or classification. The classification can be a multi-class classification (e.g., classifying a content into offensive, hate and non-hate, Davidson et al. 2017) or a binary classification (e.g., classification of real and fake news, Ahmed et al. 2017). DL algorithms which work with huge amount of data offers a significant advantage of automatically discovering the features for classification which an ML algorithm does with human intervention (Ayo et al. 2020). Considering the amount of content published on SM, neural networks have proven to be an effective tool for automatic detection of SM content.

4.2.1 Role of NLP for detection of detrimental content om SM

The manual approach of parsing the vast volume of SM text is challenging in terms of time required to understand the unstructured and noisy text, training to the moderators to parse such text which is costly. Natural Language Processing is an automated tool to parse the text using computers (Hirschberg and Manning 2015; Duarte et al. 2017). NLP has made incredible advancements in text feature representation techniques through pre-trained generalized language models. The process of converting the raw text features into numerical feature vectors is achieved using various feature representation techniques which include frequency-based techniques and neural network-based word embeddings. Scientific research articles have reported the use of these techniques in detection of detrimental content on SM. An NLP pipeline in detection of detrimental content on SM consists of Pre-processing phase and Feature Engineering phase which are detailed as follows:

Pre-processing of the content Processing and analysis of the SM data comes under the field of data and text mining. Text mining is a process of extracting knowledge and information from an unstructured and noisy data (Vijayarani et al. 2015). Processing SM content is challenging task due to the unstructured form of UGC. The UGC on social media is often noisy and written in an informal way (Ahmed et al. 2017; Robinson et al. 2018) with sentences or texts lack in punctuations, use of more abbreviations, emoticons (e.g.,::-), special characters (e.g.,"@Sush","U9","#happy") and use of repeated characters (for example "cooooll", "haaa") in the text. This ambiguous form content makes text interpretation too challenging. So pre-processing forms a crucial step to transform such free form of content into a structured form in order to have an effective analysis of the UGC. The important pre-processing steps are detailed in Table 7. As shown in Table 7 stemming and lemmatization are similar, but lemmatization is preferred over stemming as it converts each word to its base form. Stemming and lemmatization are together called as normalization (Vijayarani et al. 2015).

The pre-processing steps summarized in Table 7 vary depending on the form of the content analyzed. For example, for fake news detection, the URLs, hyperlinks are important, however for hate speech detection they may not be of much significance. The pre-processing is performed using Python NLTK library. The profane words make use special characters like "g@y", "f**c" makes tokenization challenging (Robinson et al. 2018). The pre-processing of raw text facilitates selection of features and improves the performance of ML classifiers by reducing the dimensions of input words (vocabulary words) in the text thereby reducing

the processing requirements and also selecting the features that are essential for classification.

Feature engineering Feature selection and representation together called as Feature Engineering form a noteworthy element contribute to the success of NLP text classifiers (Duarte et al. 2017). The features can be words, phrases, characters, unique words (Schmidt and Wiegand 2017; Ahmed et al. 2017) that differ depending on the form of content to be processed. The lexical, syntactic and semantic elements of text contribute to selection of features for SM content. The lexical elements are expressed at word-level lexicons in subjective, objective, formal or informal form (Verma and Srinivasan 2019). The syntactic elements refer to arrangement of words and phrases that define a sentence (Verma and Srinivasan 2019). The semantic elements include of identifying the attributes to extract the meaning of the sentence (Verma and Srinivasan 2019). The sentiments conveyed by the text can be analyzed through semantic elements. The additional features are also selected based on the meta-information accompanying the text. These include multimedia data and information about the users and its followers, geographical location which defines the environment about the content (Shu et al. 2017; Zhou et al. 2020; Fortuna and Nunes 2018; Schmidt and Wiegand 2017). In context of fake news and rumors, the lexical, semantic and syntactic features can be extracted from the news headline and main text of the news article. The images features can be extracted from image/video attribute (Shu et al. 2017; Zou and Zafarani 2020a). For hate speech content, the linguistic characteristics of the text define the features. A hate speech text is characterized by negative words (Schmidt and Wiegand 2017). An online hate message will consist of short length text, use of distinctive words that differentiate from a normal message, text with special characters, punctuation marks, user mentions etc. all from which the lexicon, syntax and semantic features can be extracted (Schmidt and Wiegand 2017; Watanabe et al. 2018; Robinson et al. 2018). The lexical, syntactic and surface features for fake news and hate speech content are similar in terms of use of words, typed dependency and use of special characters like hashtags (#), user handles (@), punctuation marks, etc. (Schmidt and Wiegand 2017; Zhang and Ghorbani 2020). For hate speech, word level features and sentiment features are considered to be important and is explored by many researchers. The use of emojis is widely used in hate speech content while the news headline forms an important feature for fake news. The feature selection method of NLP is extremely dependent on the type of SM content. The creator of news for fake news detection is used to determine the legitimate users and suspicious users (Shu et al. 2017; Zhang and Ghorbani 2020). The user profile features, user credibility features and user behavior features are deployed to determine the suspicious users which aid in detection of fake news (Zhang and Ghorbani 2020). Research has reported that the meta-information features are more important and is also exploited in detection of fake news whereas these features are considered not of much importance in hate speech detection. However, meta-information can be one of the important features for detection of certain ambiguous word content.

Feature representation is a technique of representing textual features which include words, phrases, and characters in a numerical form as shown in Fig. 7. The feature representation techniques assign a numerical value which indicates a frequency of word or a binary value which indicates the presence or absence of word in a text (Burnap et al. 2019). The numerical value represents a vector which is applied as input to ML algorithm for detection of words that are harmful. The character n-gram feature representation has shown improved performance as compared to word n- grams for noisy words like use of special characters in between the word (e.g., yrslef, a$$hole) (Schmidt and Wiegand 2017). Since Bag of Words (BoW) fail to understand polysemy word, it has shown high false positives for hate speech detection as reported in literature (Davidson et al. 2017). In some literature Parts of Speech (PoS) is considered as a pre-processing stage. BoW, n-grams and Term Frequency-Inverse Document Frequency (TF-IDF) generate the feature vectors based on frequency of words in text, there can be sparse representation of vectors due to short posts on social media which increase the memory and computational requirements. Table 8 shows the definition of the techniques. Frequency based feature representation techniques with supervised ML algorithms are used for detection of fake news, offensive content, profanity, clickbait on SM platforms. The sparse representation of feature vectors is addressed by using word embeddings. Word embeddings are pre-trained neural network based unsupervised word distribution models in which words in a huge corpus of unlabeled text are represented as numerical vectors (Schmidt and Wiegand 2017) resulting in high dimensional vector space.

The BoW technique that failed to extract the semantically similar words are addressed by word embeddings by creating the vector with values of semantically similar words placed close to each other (Mikolov et al. 2013). Research literature reports the use of word embeddings have shown a significant performance improvement in the detection of SM content using ML algorithms.

Table 9 shows the widely used word embeddings in NLP. Pre-trained word embeddings preserve the syntactic and semantic information in the text (Pennington et al. 2014). Word embedding models are trained on huge corpus with various dimensions of word vectors. In word2vec models, the pre-calculation of vectors for words serves as a limitation for words that are non-grammatical. The contextual meaning of word within the sentence is not considered in

Table 8 Feature representation techniques in NLP

Feature Representation	Description	Findings
Bag-of-Word (BoW) (Davidson et al. 2017, Robinson et al. 2018)	Each word is used as a feature with a numerical value representing the frequency of occurrence of a word in text	Fail to understand the polysemy words and fail to convey whether the word is a noun, verb or adjective
	Represent lexical features in the text	
N-grams (Davidson et al. 2017; Ahmed et al. 2017; Horne and Adali 2017)	Sequence of adjacent words or characters of length 'n' extracted from text	Fail to extract the correlations between the words that are a distance apart which would affect the performance of machine learning classifier
	The value of "n" can be 1 referred as unigrams, 2 referred as bigrams or 3 referred as trigrams	
	Character n-gram and word n-grams	
Term Frequency (TF)-Inverse Document Frequency (IDF) (Ahmed et al. 2017)	Statistical method to transform the words in a text into numeric representation	Semantic similarities between the words are not considered
	Term Frequency measures the frequency of words	
	Inverse Document Frequency indicates how important a word in a document is	
Parts-of-Speech (PoS) Tagging (Zhang and Luo 2019, Burnap et al. 2019)	Represent syntactic features that extract type dependencies by exploiting the grammatical relationships between the words in the text	Performance of machine learning classifier does not show improvement when n-gram features are combined with PoS information
	The long-distance words can be well capture with POS tagging	

Table 9 Word embeddings in NLP

Pre-trained Word Embeddings	Description	Number of words for pre-training	Dimension of vector	Findings
Word2vec by Google (Mikolov et al. 2013)	Feed forward neural network with one hidden layer	100 billion words from Google News	300	Ignore the morphological features of text and no vector representations for Out Of Vocabulary (OOV words)
	Creation of close vectors for semantically similar words			
CBOW by Google (Mikolov et al. 2013)	A continuous distributed representation that outputs a single word based on the context of neighboring words		300	Poorly utilize the statistics of the corpus and are not trained on global corpus
Skip-grams by Google (Mikolov et al. 2013)	A neural network that takes a single word as input and outputs multiple words based on the context of single word		300	
GloVe by Stanford (Pennington et al. 2015)	A log-bilinear regression model for the unsupervised learning of word representations	6B token corpus (Wikipedia 2014 + Gigaword 5)	300	Ignore the morphological features of text and no vector representations for Out Of Vocabulary (OOV words)
	Capture the global corpus statistics using co-occurrence matrix			
FastText by FaceBook AI team (Bojanowski et al. 2017)	Exploits a character level n-gram to represent a word	157 languages, trained on Common Crawl and Wikipedia	300	Long-term dependencies in the text cannot be learnt
	The generated vector of a word is the sum of its character n-grams with n ranging from 1 to 6			
EMLo by (Peter et al. 2018; Naseem et al. 2019)	Use of CNN to extract the raw characters	1 billion words with 800 M tokens of news crawl data from WMT 2011	1024	Used as augmentation with word2vec and GloVe
	Use of BiLSTM to represent context level word representations			Computationally intensive
BERT Google AI (Devlin et al. 2019, Vaswani et al. 2017)	Unsupervised language representation pre-trained on unlabeled text that considers the context of word from both sides of text	BooksCorpus (800 M words) and English Wikipedia (2500 M words)	768	Fails to work for generalized negation words
	Multi-layer bidirectional Transformer model			Computationally intensive during inference
RoBERT (Liu et al. 2019)	Similar to BERT	160 GB Text (16 GB of BERT + CommonCrawl News dataset (63 million articles, 76 GB), Web text corpus (38 GB) & Stories from Common Crawl (31 GB))	–	More training time than BERT
	Large batch training size			
ALBERT (Lan et al. 2019)	18 × fewer number of parameters as compared to BERT	English Wikipedia (2,500 M words) + Stories from Common Crawl (31 GB))	128	Reduced accuracy
	Reduced size of Embedding layer			Computationally intensive
	Parameter sharing achieved through one encoder layer applied each time on input			

Table 9 (continued)

Pre-trained Word Embeddings	Description	Number of words for pre-training	Dimension of vector	Findings
DistilBERT (Sanh et al. 2020)	40% less parameters than BERT	16 GB of BERT + 3.3 billion words	–	Suffer from biased predictions
	Use of knowledge distillation to red			
XLNET (Yang et al. 2020)	Auto-regressive language model	130 GB of textual data (33 billion words)	–	More Computationally and resource intensive
	Permutation Language modeling with all tokens predicted in random order			Longer training time than BERT
GPT-3 (Brown et al. 2020)	175 billion parameters	Common Crawl (410 billion tokens), webtexts (19 billion tokens), books (27 billion tokens), and Wikipedia (3 billion tokens)	12,888	Complex and costly inference
				Text generated by the model can introduce bias in language

CBOW Continuous Bag Of Words, *GloVe* Global Vectors for word representation, *BERT* Bidirectional Encoder Representations from Transformers, *EMLo* Embeddings from Language Models. *RoBERTa* A Robustly Optimized BERT Pretraining Approach. *ALBERT* A Lite BERT Self-supervised Learning of Language Representations, *GPT* Generative Pre-trained Transformer

word2vec model. This contextual understanding is considered in BERT and EMLo in which the vectors are calculated depending upon the context of word in the sentence. The real time calculation of vector representations has shown significant results in terms of accuracy in detection of SM content as reported in literature. BERT and EMLo are deep bidirectional language models that work on transfer learning (Pan and Yang 2010) concept are pre-trained on a corpus and are fine-tuned for a new corpus (Devlin et al. 2019). Both CBOW and skip-gram exhibit low computational complexity and can be trained on a large dataset; however, BERT and EMLo are computationally intensive indicating more response time. The feature vectors are applied as an input to a ML algorithm or a DL algorithm. As shown in Table 9, word embeddings are self-supervised pre-trained language models that are trained on large unlabeled dataset. Considering the amount of data (from 100 billion words to 130 GB of text data) the language models are trained on implies an increased number of hyperparameters (3000 of Word2Vec to 175 billion of GPT-3). This also signifies an increased training time to train the model and the number of computational resources required for training. For example, XLNet requires 512 TPUs and 2.5 days for training (Yang et al. 2020). Pre-trained language models are experimented for detection of fake news and hate speech on SM. Table 10 shows the use of language models for detection of detrimental content. As shown in Table 10, pre-trained language models perform better for fake news detection task and have reported low F1-score for hate speech detection task. However, BERT pre-trained on COVID-19 fake news dataset extracted from Twitter has reported highest F1-score.

This indicates that there is a need to create pre-trained language model that will consists of words and phrases that target on inflammatory or abusive words. The skewed nature of datasets also affects the performance of pre-trained language models. Malik et al. 2022 have experimented transformer models like small BERT (trained on less amount of dataset), BERT, ALBERT on three different datasets of hate speech and offensive language and compared the performance of these language models in terms of training time the model takes per epoch. The study reported that the training time of ALBERT language model is highest as compared to BERT and small BERT but ALBERT performed better in terms of F1-score (90%) than other models. The computational efficiency of language model in terms of training time is a crucial factor that needs to be considered for detection of detrimental content on SM.

4.2.2 ML and DL algorithms for detection of detrimental content on SM platforms

ML is a vital and largest subfield of AI that includes techniques to provide systems the ability to automatically

Table 10 Language models for detection of detrimental on SM

Type of detrimental content	Dataset	Language Model	Performance metric	Findings
Fake News Glazkova et al	COVID-19 Healthcare Misinformation Dataset (Cui and Lee 2020)	BERT	F1- score: 96.75%	Misclassification of true posts about corona vaccine predicted false by the model
		RoBERT	F1- score: 97.62%	
		COVID-Twitter-BERT (CT-BERT)	F1- score: 98.37%	
Hate Speech (Zhou et al. 2020)	SemEval 2019 Task 5	EMLo	F1- score: 63.6%	Fusion method resulted in better F1-score
		$BERT_{base}$	F1- score: 62.3%	
		CNN	F1- score: 69.8%	
		Mean Fusion method of EMLo + BERT + CNN	F1- score: 70.4%	
Hate Speech and Offensive Language (Mutanga et al. 2020)	(Davidson et al. 2017)	BERT	F1- score: 73%	DistilBERT outperformed other transformer models
		RoBERT	F1- score: 69%	
		LSTM with attention	F1- score: 66%	
		XLNET	F1- score: 72%	
		DistilBERT	F1- score: 75%	
Satire news (Li et al. 2020)	4000 satirical and 6000 regular news articles	Vision & Language BERT (ViLBERT) (Lu et al. 2019)	F1- score: 92.16%	Model fails to extract relationship between text and image which results into misclassification

learn and improve from experience without being explicitly programmed. Many subfields of AI are addressed with ML methods (Ofcom 209). Figure 8 shows the process of detection and classification of a SM content using ML algorithms. Research literature have reported the use of supervised ML algorithms like SVM, LR, NB, and RF for the detection and classification of SM content which predominantly include fake news and hate speech. The ML algorithms are trained on various features extracted from the labeled datasets using BoW, TF-IDF, n-grams feature representation techniques. The trained ML algorithm is applied on the test data for classification as shown in Fig. 8. The classification can be a multiclass classification for example classifying a content into offensive, hate and non-hate (Davidson et al. 2017) or a binary classification for example classification of real and fake news (Ahmed et al. 2017). The performance of ML algorithm is evaluated on the datasets which contain a huge data extracted from popular SM platforms like Facebook, Twitter, Instagram, and Reddit. ML algorithms are considered as traditional algorithms for detection and of SM content. The handcrafted features used by ML algorithms are time consuming, incomplete, and labor intensive with the performance of a ML algorithm is dependent on the features selected for classification. Deep Learning (DL) a sub-field of ML has attracted the industry and academia for various applications. DL is basically a neural network with an input layer, one or more hidden layers and an output layer (Ayo et al. 2020).

A neural network with more hidden layers is a deep neural network. DL algorithms make use of deep neural networks to train on a data and predict the output or do classification. DL which works with huge amount of data offers a significant advantage of automatically discovering the features for classification which an ML algorithm does with human intervention (Ayo et al. 2020). Considering the amount of content published on SM, neural networks have been an effective tool for automatic detection of SM content. Table 11 depicts the various neural network models deployed for detection of SM content. Considering the various characteristics of SM content, different neural network models are deployed. For example, discriminative models that consider SM content and context features are CNN and RNN (Islam et al. 2020).

Generative models like Generative Adversarial Network (GAN) and (VAE) that generate new data are explored for rumor detection (Ma et al. 2019; Sahu et al. 2019; Khattar et al. 2019). Hybrid models like CNN-RNN, RNN-GRU, CNN-LSTM, GAN-RNN (Shu et al. 2019; Badjatiya et al. 2017; Zhang et al. 2018) are explored for multimodal approach of SM content detection with visual features and textual features from two neural networks concatenated together for a classification task.

The performance of a machine learning and neural network applied to a particular task is evaluated on the performance metrics detailed in Table 12. The detection of SM content using ML algorithms and DL is evaluated using accuracy, precision recall and F1-score. The performance of ML algorithm is tested for the number of false positives and false negatives which implies the misclassification rate of a specific content. For example, non-hate speech content misclassified as a hate content which indicates a high false negative. It is desirable that the ML algorithm should

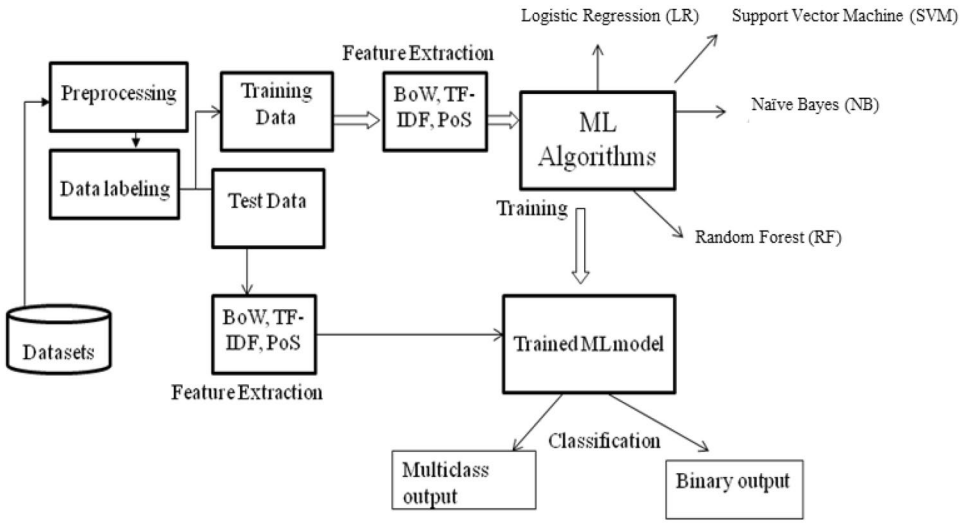


Fig. 8 Process of detection and classification of a SM content using ML algorithms

Table 11 Neural network model for SM content detection and classification

Deep Neural Network model	Description
CNN (Ayo et al. 2017, Islam et al. 2020, Gambäck and Sikdar 2017)	Trained with word vectors using a fixed kernel size and number of filters
	1-Dimensional CNN (1D-CNN) is used to extract the local features from the text
	Extract the image features in multimodal approach of SM content detection
RNN (Nasir et al. 2021)	Sequential neural network with internal memory to process the short text
	For fake news detection, RNN is used to capture the temporal features of posts over time
LSTM (Ayo et al. 2017, Ruchansky et al. 2017)	Special type of RNN that learns the typed dependencies in the long text
	The memory unit consists of cell that use gates and a carry
	Carry responsible retaining the information during the sequential process
GRU (Ayo et al. 2017, Amrutha and Bindu 2019)	Addresses the short-term memory problem of RNN with gates
	Gates that decide the amount of information to be passed in the network and amount of information to be neglected
HAN (Singhania et al. 2017, Singh et al. 2020)	Makes use of an attention layer between the encoder LSTM and decoder LSTM
	Attention is given to each encoder input at each time step
GAN (Ma et al. 2019, Sahu et al. 2019, Islam et al. 2020)	An unsupervised learning method that generates new data
	Discriminative model used for classification
VAE (Khattar et al. 2019, Islam et al. 2020)	A generative autoencoder model that learns the latent state distribution of input data in a probabilistic manner
	Consists of encoder, decoder and a loss function
Capsule networks (Goldani et al. 2020)	Capsule indicates a group of neurons
	Text features extracted through n-gram convolutional layer
	Features are then processed in primary capsule layer, convolutional capsule layer and feed forward capsule layer
Transfer Learning (Pan and Yang 2010)	A knowledge transfer ML model in which learned features trained on one task are reused for learning another task through language models like BERT, RoBERT

achieve a low false negative rate. Automated Techniques for fake news detection rely on AI based techniques with NLP tools combined with traditional ML algorithms and DL techniques. Various research articles have reported detection and classification of fake news and its types by exploring its content, user and social network characteristics (Shu

Table 12 Performance metrics of ML algorithm and DL

Performance Metric	Table/Formula
Confusion Matrix: The complete performance of a ML algorithm is measured with the confusion matrix with 2×2 dimensions i.e., "Actual" and "Predicted" giving an output with 4 values: "True Positive (TP)", "False Negative (FP)", "False Positive (FP)", and "True Negative (TN)"	*Predicted* / *Actual*: *True Positive(TP)*, *False Positive (FP)*, *False Negative (FN)*, *True Negative (TN)*
Accuracy (A): It is a common evaluation metric that measures the correct prediction made by an algorithm	$Accuracy = \frac{TP+TN}{TP+FN+FP+FN}$
Precision (P): It is the number of correct positive results divided by the number of positive results predicted by the algorithm	$Precision = \frac{TP}{TP+FP}$
Recall or Sensitivity (R): It refers to the true positive rate and summarizes how well the positive class was predicted by an algorithm	$Recall = \frac{TP}{TP+FN}$
Specificity: It refers to the true negative rate and summarizes how well the negative class was predicted by an algorithm	$Specificity = \frac{TN}{FP+TN}$
F1-Score(F1): It is a harmonic mean between precision and recall and measures the robustness of an algorithm	$F1-score = 2 \times \frac{Precision \times Recall}{Precision+Recall}$

et al. 2017; Zhou and Zafarani 2020). Various supervised ML algorithms like NB, SVM, KNN, LR, DT, and RF are experimented on various datasets to classify fake news as a binary classification task or a multi-class classification task.

Table 13 depicts the various supervised ML algorithms for detection of different forms of fake news like satire news, rumors, clickbait. Table 13 shows the diversity in features and also the datasets for detection of various forms of fake news. In context of fake news, detection involves classifying a piece of information as real or false which can be considered as two class classification problem. Most of the research literature shows the use of supervised ML algorithms that work on the available datasets for detection. There is a need to exploit unsupervised ML algorithms for detection. The lexical features, semantic and syntactic features are common feature selection methods for all forms of fake news. The writing style-based features vary for rumors, satire and clickbait detection. Many researchers have considered accuracy as a performance metric for evaluating the ML algorithm, however precision and recall are also important metrics that provide the percentage of fake news detected. The labor intensive and time-consuming task of developing handcrafted features for ML algorithms is considered by deploying DL neural networks which process huge amount of data without human intervention for extracting the features from such data.

Table 14 shows the use of supervised ML algorithms and ensemble ML algorithms (Malmasi and Zampieri 2018) for hate speech detection with SVM and LR reported better performance. In ensemble classifiers, individual classifiers are combined using various methods like Borda Count, Mean Probability Rule, and Median Probability Rule which help in improving the accuracy of classification task (Malmasi and Zampieri 2018). The ML algorithms are experimented on datasets and these datasets include diverse and fine-grained form of hate speech content. Twitter is widely used platforms for accessing the hate speech forms and creating a dataset. Figure 9 shows the statistics of ML algorithms deployed for detection and classification of SM content. As shown in Fig. 9 SVM is the most widely used algorithm for SM content detection with average accuracy of around 75% to 80%. SVM algorithm has shown increased accuracy for fake news detection as compared to hate speech. This is due to subjectiveness and variations in the hate speech words whereas fake news is objective in nature.

The performance metrics of an ML algorithm are more dependent on the datasets on which it is experimented. The ability to process huge data with automatic extraction of features from the data is unique characteristic of DL neural network models. This characteristic is explored in form of extracting various features like news content features, user responses to news, temporal characteristics using social graph which aid in fake news detection by neural network models. The ML algorithms perform best for small datasets with TF-IDF representation technique (Cunha et al. 2021).

As shown in Table 15, pre-trained word embeddings are the most common feature representation techniques for classification. CNN and GAN architectures have shown significant performance in NLP tasks like text classification, sentiment analysis (Goldani et al. 2020). Transformer models have reported better classification accuracy for large datasets but at the cost of increased computational time and resources (Cunha et al. 2021).

State-of-art hybrid architectures like CNN-RNN, Attention- LSTM have also reported promising results in terms of accuracy and F1-score with few architectures implementing early detection of rumors and fake news. However, the time indication of early detection is not addressed in the literature.

Table 13 ML algorithms for various forms of fake news detection

Type of fake news	Dataset	Feature selection	ML algorithm	Performance Metric (in %)				Findings
				A	P	R	F1	
Fake news (Granik and Mesyura 2017)	BuzzFeed	Text features	NB	74	–	–	–	Low value of recall
Satire and humor News (Rubin et al. 2016)	360 news articles	Absurdity, Humor, Grammar, Negative Affect, Punctuation TF-IDF	SVM		90	84	87	Punctuation marks features are important for satire detection
Fake news (Ahmed et al. 2017)	Kaggle	Text features N-gram with TF-IDF	SVM	86	–	–	–	Best accuracy obtained for unigram features with decreased accuracy for n=4
			LSVM	92	–	–	–	
			KNN	83	–	–	–	
			SGD	89	–	–	–	
			DT	89	–	–	–	
			LR	89	–	–	–	
	BuzzFeed	N-gram with TF-IDF	LSVM	87	–	–	–	
Fake news (Horne and Adali 2017)	BuzzFeed	number of nouns, lexical redundancy (TTR), word count, number of quotes	LSVM	71	–	–	–	High accuracy for fake and satire news from real news
Satire news (Horne and Adali 2017)	Burfoot and Baldwin 2009		LSVM	67	–	–	–	
Real, fake and satire (Horne and Adali 2017)	Political news dataset		LSVM	91	–	–	–	
Fake news (Kleinberg et al. 2017)	Fake news AMT	Lexical, syntactic, semantic, readability features	LSVM	60	–	–	57	Comparative analysis of manual and automatic fake news detection
	Celebrity Fake news			61	–	–	57	
Rumors (Zhang et al. 2015)	Sina Weibo	Shallow text features, implicit user and content features	SVM	–	72	59	65	Exploration of rumor characteristics for detection
Rumor (Yang et al. 2015)	Twitter	Content based features, Twitter based features and Network	LR	–	98	95	96	Rumor identification using hot topic detection
			NB	–	98	90	94	
			RF	–	98	99	98	
Clickbait (Chakraborty et al. 2015)	7623 articles from BuzzFeed, Viralnova, Scoopwhoop, Viral stories	Sentence structure, word patters, n-grams, clickbait language	SVM	93	95	90	93	Work on English headlines
			DT	90	91	89	90	Approaches to block clickbaits
			RF	92	94	91	92	

Table 14 ML algorithms for various forms of hate speech detection

Type of Hate Speech	Data Source	Feature extraction	ML classifier	Performance Metrics (%)			Findings
				P	R	F1	
Hateful and antagonistic content (Burnap and Williams 2015)	Twitter	n-gram	BLR	89	69	77	Syntactic features reduced false negatives by 7%
		BOW	RFDT	89	68	77	
			SVM	89	68	77	
Hateful Offensive (Davidson et al. 2017)	Twitter	Bigram	LR	91	90	90	Multi-class classification
		Unigram					40% of hate speech misclassified
		Trigram					
		Each weighted by its TF-IDF					
Hate and Offensive (Watanabe et al. 2018)	Twitter	Sentiment based	RF	60	59	59	Binary classification for clean and Offensive text
		Semantic					
		Pattern	SVM	64	57	60	
		Unigram	J48graft-DT	79	78	78	Ternary classification for hate speech
Abusive language (Nobata et al. 2016)	Yahoo	Token n-grams	LR	77	79	78	Context of comment not taken into account
		Characters n-grams					
		Word2vec					
Aggression (Modha et al. 2020)	TRAC-Facebook	Unigrams with tf-idf	LR	68	57	60	Multiclass classification of aggression as overtly aggressive, covertly aggressive, non-aggressive
		Char 5-g					
		Length of the post	SVM	68	57	60	
	TRAC-Twitter		LR	52	52	49	Better performance for Facebook dataset
			SVM	49	49	49	
Racist (Kwok and Wang 2013)	Twitter	Unigrams	NB	**A**: 76			Considered only hate speech against blacks
							Reduced accuracy outside the context
Cyberbullying text (Dinaker et al. 2011)	Youtube comments	TF-IDF	NB	**A**: 63			Clustering of messages relevant to cyberbullying
		POS	J48 DT	**A**: 61			Detection of profanity and negativity from the clusters
			SVM	**A**: 66			
Hate, Aggressive, Profanity (Sharma et al. 2018)	Twitter	TF-IDF	NB	**A**:73.42			Harmful Speech categorized into 3 classes
		BOW	RF	**A**:76.42			
		TF-IDF	SVM	**A**:71.71			
Hate, Offensive (Malmasi and Zampieri 2018)	(Davidson et al. 2017)	Surface n-grams	Ensemble Classifier	**A**: 77			Misclassification of hate class as offensive
		Word skip grams					
		Brown clusters	LSVM	**A**: 78			
			RBF-SVM	**A**: 80			

The social context for fake news detection task is also considered by neural network models like in CSI architecture (Ruchansky et al. 2017) and FANG (Nguyen et al. 2020) architecture. Nguyen et al. (2020) reported a Factual News Graph (FANG) framework that constructs a social context graph of list of news articles, news sources, social users and social interactions using Graph Neural Networks. The FANG framework showed AUC of 0.7518 on limited training data. However, the depending on the event for which fake news and rumors are disseminated, the social network graph features will change indicating the importance context that needs to be taken care for real time detection of fake news and rumors. The DL algorithms based on neural network architectures have outperformed traditional ML algorithms for detection of hate speech task. Various DL techniques like CNN, RNN, LSTM, Capsule networks, and

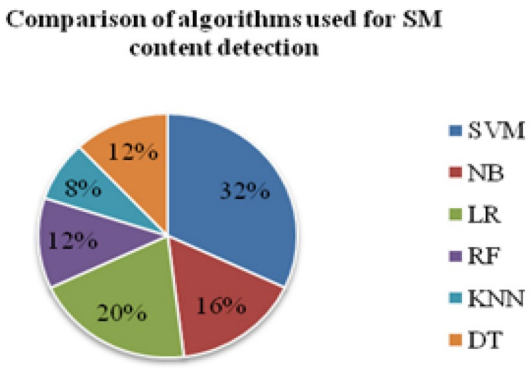


Fig. 9 Statistics of ML algorithms for SM content Detection

Transformer models have shown good performance in terms of accuracy and F1-score.

Hybrid architecture like VAE + CNN (Qian et al. 2018) are experimented to generate user responses to news articles and extract semantic text features from posts assist in early detection of fake news.

Table 16 summarizes the DL techniques for detection of hate speech as reported in research. As shown in Table 16, various state-of-art DL techniques with hybrid neural networks are deployed for hate speech detection. CNN architectures are able to extract contextual features which are exploited in form of character CNN and word CNN (Park and Fung 2017) for hate speech detection and sentence level features with margin loss for fake news detection (Goldani et al. 2020). Like traditional ML algorithms, DL techniques also fail to detect and classify fine grained hate speech content like abusive content, offensive content and aggressive content. The error analysis for detection and classification of such content is missing and needs to be considered in research.

4.2.3 Multimodal approach of detecting detrimental content on SM

The multimedia forms an important attribute and modality that can assist in the moderation of SM content. The multimedia content includes images, videos, GIFs (Graphics Interchange Format). The development in multimedia technology has shifted the paradigm of text-based news articles to news articles that include a images and videos accompanied with text which attracts a greater number of readers (Qi et al. 2019). For example, a post or a tweet with images gets 89% more likes and a number of reposts for a tweet or posts with images is 11 times larger than a post without image (Cao et al. 2020). Recent years has also observed a rise in fake images attached to news article. As reported by Qi et al. (2019), the visual content which are false can be in form of tampered images, misleading images and images with wrong claim as shown in Fig. 10 (Qi et al. 2019, 2020). For detection of fake news from visual content includes exploring the diverse characteristics of the fake image (Cao et al. 2020) as these characteristics differ from a real image. These characteristics form the features which include forensic features, time-context features and statistical features (Cao et al. 2020) that are extracted to determine correctness of image. Qi et al. (Dinakar et al. 2011) experimented with forensic features using DCT to transform the image from pixel domain to frequency domain and the multiple semantic features of the image were captured using CNN with a bidirectional GRU (Bi-GRU) network to model the sequential dependencies between these features. These two features were concatenated together to detect the fake news achieving a accuracy of 84.6%. Boididou et al. (2015) experimented with forensic features and extracted descriptive statistics to detect fake news. The forensic features were combined with content-based features and user-based features which showed recall of 0.749, precision of 0.994 and F1- score of 0.854. The capabilities of DL neural networks are extended by combining the content and visual features together for detection of fake news and have shown promising results in terms of early detection of fake news and event discriminators (Wang et al. 2018). The user on a social media network publishes the content using different modalities like text, image and video. This form of modality is also observed in fake news and hate speech content sharing on social media. Majority of research literature has focused on exploring the textual content of news article for fake news detection. A textual content accompanied with visual content conveys more information that will assists in detection process.

Combining the textual and visual features together is challenging in terms of different characteristics like complex and noisy pattern of news article. Research studies have reported state-of-the-art multimodal architectures that are detailed next.

Figures 11 and 12 illustrate the multimodal approach of fake news detection. The two architectures concatenated the textual and visual representation features for detection of fake news. SpotFake (Singhal et al. 2019) architecture experimented with visual and textual features to classify fake news and EANN architecture explored the multimodal features for fake news detection and capturing event invariant features. EANN architecture (Wang et al. 2018) extracted the textual and visual features using CNN while BERT language model is used to extract the text features from the news articles in SpotFake architecture. Table 17 presents the multimodal architectures for fake news and rumor detection. The visual content extraction is done using VGG-19 (convolutional network pre-trained on ImageNet dataset) by most of the architectures. The dimensional vectors of visual and text modalities are made similar and combined together to

Table 15 DL neural network models for fake news detection

Refs	Feature extraction method	Features of DL neural network used for detection	Dataset	Performance metric (in %)
Nasir et al. (2021)	Word2vec	Hybrid CNN-RNN architecture	ISOT	A, P, R, F1: 99
		Training on ISOT dataset and testing on FA-KES dataset	FAKES	A, P, R, F1: 60
Singhania et al. (2017)	GloVe	Representation of news article with news vectors	20,372 articles from 16 sites labeled fake	A: 96.24 for 3HAN
		3 level HAN for word, sentence and headline of input article		
		Bidirectional GRU for word, sentence and headline encoder	20,932 articles from 9 websites labeled genuine	A: 96.77 for Pre-trained 3HAN
		Attention weights to word, sentence and headline through heatmap		
Goldani et al. (2020)	n-grams	Pre-trained static, non-static and multi-channel word embeddings	ISOT	A: 99.8
		Model includes n-gram convolutional layer, the primary capsule layer convolutional capsule layer, and a feed-forward capsule layer for different length news statement	LIAR	
Paka et al. (2020)	BERT word embeddings	Creation of COVID-19 misinformation dataset	CTF (**C**OVID-19 **T**witter **F**ake News)	F1: 95
		Cross-Sean a semi-supervised attention neural model on unlabeled tweet texts		
		Encoding textual data using bidirectional LSTM		
		Cross-stitch unit for encoding user and tweet features		
		Chrome-SEAN, a chrome extension to flag COVID-19 fake news on Twitter		
Momtazi et al. (2020)	Static word embedding	Embedding layer with pre-trained word vectors	ISOT	ISOT: A: 99.1
	Non-Static word embedding	CNN layer with 3-g features of sentence	LIAR	
	Multi-channel embedding	Fully connected layer for classification using margin loss and softmax function		

Table 15 (continued)

Refs	Feature extraction method	Features of DL neural network used for detection	Dataset	Performance metric (in %)
Shu et al. (2019)	News sentences and user comments	Weighted sum of attention vectors for news sentence features and user comments features	GossipCop Politifact	GossipCop: A:80.8 P: 72.9 R: 72.2 F1-: 75.5
		Explainable fake news detection		
	Word and sentence level encoding with bidirectional RNN with GRU	Use of metric MAP@k(Mean Average Precision)(k = 5 or 10) to evaluate the model		Politifact: A:90.4 P: 90.2 R: 95.6 F1: 92.8
Qian et al. (2018)	Sentence and word representation	Early detection of fake news Two Level CNN to capture semantic information from long text Conditional Variational Autoencoder (CVAE) to generate user responses conditioned to a given news article	Weibo Twitter	Weibo: A: 89.84 Twitter: A:88.83
Dong et al. (2019)	Hybrid feature learning unit based on RNN	Credibility inference model from heterogeneous information fusion within the social networks A gated diffusive unit model that exploits the relationship among news articles, creators and subjects Hybrid feature learning unit for textual content and explicit and latent features	Politifact	A: 63
Hamdi et al. (2020)	Node2vec	Hybrid approach that exploits the credibility of information sources Graph embeddings to extract features from Twitter social graph User features combined with user graph features	Graph Dataset: ego-Twitter CREDBANK	A: 98.21 P: 91.3 R:99 F1: 98.2
Ruchansky et al. (2017)	Doc2vec	RNN to capture the temporal patterns of text LSTM model for temporal response of users to the article Users source characteristics using weighted user graph Prediction using the combined temporal and textual features and source score of users	Twitter Weibo	Twitter: A: 89.2 F1: 89.4 Weibo: A: 95.3 F1: 95.4

Table 15 (continued)

Refs	Feature extraction method	Features of DL neural network used for detection	Dataset	Performance metric (in %)
Ma et al. (2019)	Textual features using BoW	Rumor detection using generator and discriminative model (GAN)	Twitter PHEME	Twitter: A: 86, P: 88 R: 89 F1: 86 PHEME: A: 78, P: 77 R: 79 F1:78
		Generator model trained on a claim to generate uncertain and conflicting issues		
		Discriminative classifier learns to distinguish whether an instance is from real world using discriminative and low frequency patterns		
Singh et al. (2020)	13 linguistic and user features using GloVe	Hybrid feature extraction using CNN and LSTM	PHEME	P: 82, R: 81 F1: 82
		Selection of optimal feature set using Particle Swarm Optimization (PSO) algorithm		
		Attention LSTM for rumor veracity detection		

form a feature vector which is then applied to a fully connected neural network with hidden layers and a classification layer. There is need to utilize these architectures for real time detection of fake news.

As reported in many research articles a single modality feature is not sufficient to identify a hate speech or abusive content. Many ML algorithms have reported false positive rates as certain words are either misclassified as hate speech words. A user on a social media can use various modalities like text, video, image and audio to share. The image accompanied with text will assist in detection of hate speech. Research studies have reported the use of image and text modalities for detection of hate speech. Kumar et al. (2021) presented a multi-modal neural network-based model that combined the text and image features to classify asocial media post into Racist, Sexist, Homophobic, Religion-based hate, other hate and No hate. Figure 13 shows the neural network model architecture for hate speech classification as reported in Kumar et al. (2021). The image content features are extracted using pre-trained CNN based VGG-16 network. The text features are extracted using text CNN architecture with GloVe word embeddings. The text and image features are concatenated and then applied to softmax layer for classification into 6 classes of hate speech. The model achieved weighted precision of 82%, weighted recall of 83%, and weighted F1- scores of 81% tested on the Dataset MMHS150K. The proposed model achieves high true positives for non-hate class with high false positive for homophobe and religion class. Kumari et al. (2021) reported a multimodal approach for a multiclass classification of cyber-aggression on social media for posts which consists of symbolic image together with text.

The symbolic image features were extracted using VGG-16 network and textual features using CNN with three layers. The concatenated image and textual features were optimized using Binary Particle Swarm Optimization (BPSO) algorithm.

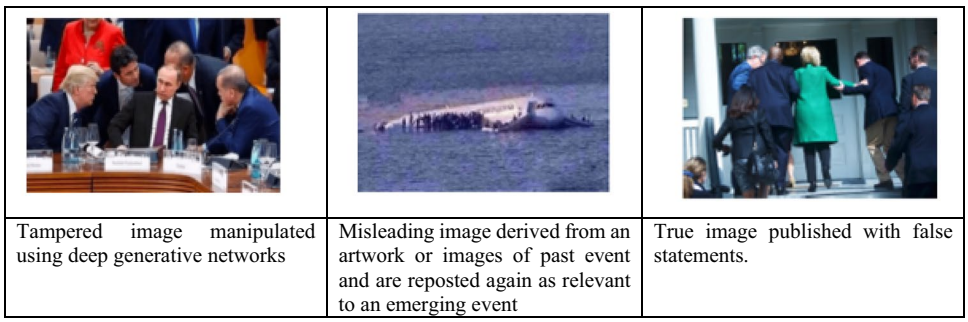
Using BPSO algorithm, the redundant features were eliminated, and the new hybrid features were applied to Random Forest ML classifier to classify the social media posts into non-aggressive, medium- aggressive and high aggressive. The dimensions of concatenated features were reduced from 1024 to 507 using BPSO algorithm. The proposed system achieved weighted precision of 74%, weighted recall of 75%, and weighted F1-scores of 75% on a created dataset of 3600 images together with text acquired from Facebook, Twitter and Instagram. The study reported to have a performance

Table 16 Deep learning techniques for hate speech detection

Type of hate speech	Data source	Feature extraction	ML classifier	Performance metrics (%)				Findings
				P	**R**	**F1**	**A**	Random Embeddings (RE) detect hatred words better than GloVe embedding
Sexist, Racist (Badjatiya et al. 2017)	Twitter	Glove word embedding	LSTM + RE + GBDT	93	93	93	–	
			CNN + RE + GBDT	86	86	86	–	
			FastText + RE + GBDT	88	88	88	–	
Aggression (Modha et al. 2020)	TRAC-Twitter	FastText	CNN	70	60	64	–	Covertly aggressive content misclassified as non-aggressive
			BiLSTM + attention	71	51	55		
			BERT	72	58	62		Real time visualization of aggressive comments through web-browser plugin
	TRAC-Facebook		CNN	57	59	55	–	
			BiLSTM + attention	56	58	58		
			BERT	58	58	57		
Hate words (Amrutha and Bindu 2019)	Twitter	Word Embeddings	GRU	–	–	65	95	Models evaluated with 2 performance metrics
			CNN	–	–	64	94	
			ULMFiT			97	97	
								Context around the hate word not considered
Cyber hate: Religion (Liu et al. 2019)	Twitter	BOW	Fuzzy based: 4 fuzzy forms + KNN	84	40	5267	-	Fusion of multiple fuzzy classifiers and instance-based reasoning to detect the ambiguous instances in different types of hate speech
		Doc2vec						
Race				93	50	46	–	
Disability				96	60	74	–	
Sexual orientation				69	35	46	–	
Abusive language (Park and Fung 2017)	Twitter	Word2vec	CharCNN	74	67	70	-	Multi class classification of abusive and non-abusive content
			WordCNN	73	72	73		
			HybridCNN	72	75	73		
Sexist and racist comments		Word2vec	CharCNN + LR	94	94	94	–	Sexist and racist classification using CNN with LR
			Word CNN + LR	95	95	95	–	
			Hybrid CNN + LR	95	95	95	–	
Racism, sexism (Gambäck and Sikdar 2017)	Twitter	Random vector	CNN	87	67	75	–	Multi-level classification based on feature embedding
		Word2vec		86	72	78	–	
		Character n-gram		85	70	76	–	High false positive rate for non-hate text
		Word2vec + character n-gram		86	70	77	–	Not able to identify combined racist and sexist content
Sexist and Racist Comments (Pitsilis et al. 2018)	Twitter	Word embedding	RNN-LSTM	93	93	93	–	Multi class classification of sexist, racist and neutral content
								User behavior to content considered as feature
Hate speech (Roy et al. 2020)	Twitter	GloVe	C-LSTM	75	43	55	–	tenfold cross validation gave best recall prediction for hate and non-hate classification
			LSTM	64	53	58	–	
			DCNN	97	88	92	–	

Table 16 (continued)

Type of hate speech	Data source	Feature extraction	ML classifier	Performance metrics (%)				Findings
				P	**R**	**F1**	**A**	Random Embeddings (RE) detect hatred words better than GloVe embedding
Hate Inducing Abusive (Mathur et al. 2018)	HOET	GloVe	Ternary Trans-CNN	80	69	71	84	Three class classification for code switched hate speech Transfer Learning neural network model



Tampered image manipulated using deep generative networks

Misleading image derived from an artwork or images of past event and are reposted again as relevant to an emerging event

True image published with false statements.

Fig. 10 Example images in fake news articles

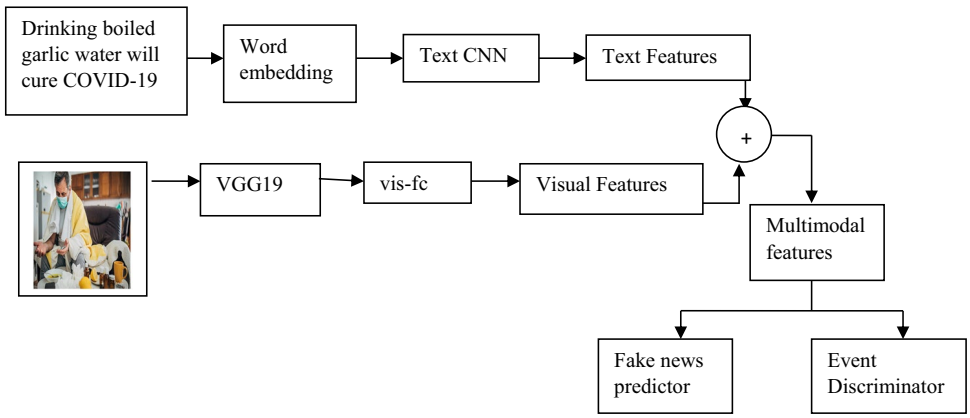


Fig. 11 Architecture of EANN: Event Adversarial Neural Networks for Multi-Modal Fake News Detection (Wang et al. 2018)

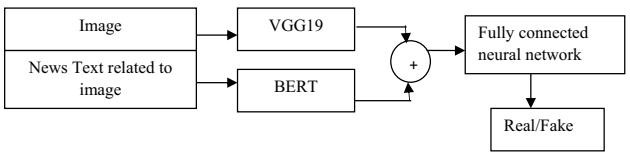


Fig. 12 Architecture of SpotFake (Singhal et al. 2019)

improvement of 3% when optimized features are used for classification.

Cheng et al. (2019) reported a collaborative multimodal approach of cyberbullying detection based on heterogeneous network representation learning. The study used 5 modalities from Instagram like image, user profile (number of followers, the total number of comments, and the total number of likes received), timestamp of posting an image, description of the image and comments, and dependencies between social media sessions through relations among users. The system reported a Macro F1-score of 96% and Micro F1-score of 98%.

Sahu et al. (2021) experimented with GAN-fusion model that combined different adversarial models for text, caption and image achieving a precision of 61%, recall of 51% and F1-score of 56%. The experimentation was done on a MMHS150K dataset which includes image, its caption and text.

Table 17 Multimodal architectures for fake news detection

Refs	Text representation	Visual content representation	Performance metric				Dataset	Key findings/Features
			A	P	R	F1		
(Wang et al. 2018)	Text-CNN	Pre-trained VGG-19	72	82	64	72	Twitter	Multimodal features to learn the discriminative
			83	85	81	83	Weibo	Representations for fake news identification learn the event invariant representations by removing the event-specific features
(Singhal et al. 2019)	BERT	Pre-trained VGG-19	77	75	90	82	Twitter for fake news	Explored only content and visual features for detection. Empirical analysis of fake news through public surveys
			89	90	96	93	Weibo for fake news	
Khattar et al. (2019)	Bi-LSTM	Pre-trained VGG-19	75	80	72	76	Twitter for fake news	Variational Autoencoder to discover the correlations of text and visual features for fake news detection
			82	85	77	81	Weibo for fake news	
Zafarani et al. (2020b)	Text-CNN	Image2sentence model	87	88	90	89	Politifact	Recognizing the falsity of news by detecting the mismatch between image and text
			83	85	93	89	GossipCop	
Cui et al. (2019)	Glove for content	VGG	Micro F1 for 80% training ratio: 77				Politifact	Latent sentiments in the users' comments for fake news detection
	One- hot encoding for profile and user comments		Macro F1 for 80% training ratio:76				GossipCop	Adversarial mechanism to conserve semantic relevance and the representation consistency across different modalities
			Micro F1 for 80% training ratio: 80					
			Macro F1 for 80% training ratio:81					

The multimodal approach of various forms of hate speech detection includes extracting features from different modalities using deep neural networks. For multi-modal detection, context is important feature which is missing in reported systems and needs to be considered in research. The method of concatenating the features from different modalities is less detailed in the literature.

5 Moderation of detrimental content on SM platforms

The exploitation of SM for a wrong purpose is increasing substantially every year and is imposing challenges to various sectors like private organizations, government and civil society (Ganesh and Jonathan 2020). Inspite of legal measures enforced by the government to control the devastating detrimental content on SM, the dissemination of such content has not stopped. So, content detection and moderation on SM platforms is of primary importance. Content moderation on online platforms has drawn attention in academia with many research articles published in scientific journals. Traditional publishing platforms detect and moderate the content by verifying the content with known facts (Wyrwoll 2014). Content moderation involves decisions about decreasing the presence of extremist contents or suspending exponents of extremist viewpoints on a platform (Ganesh and Jonathan 2020), elimination of offensive or insulting material, the deletion or removal of posts, the banning of users (by username or IP address), making use of text filters

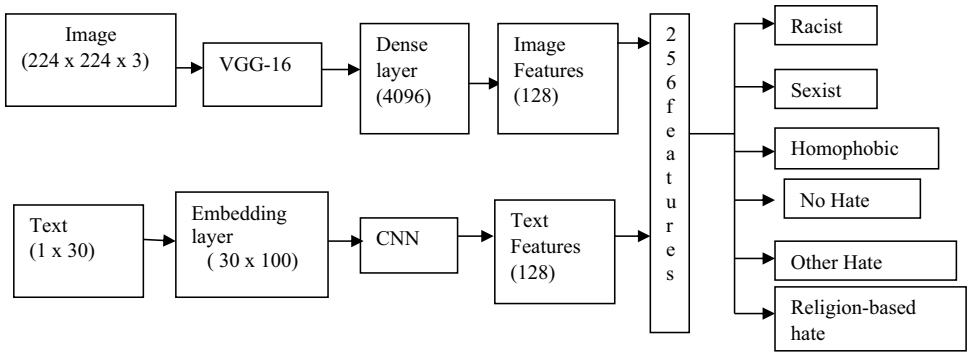


Fig. 13 Neural network model architecture for multimodal hate speech classification Kumar et al. (2021)

to disallow posting of specific types of words or content, and other explicit moderation actions (Ganesh and Jonathan 2020). Content moderation involves law enforcement organizations set by government and civil society (Ganesh and Jonathan 2020). Commercial content moderation is a method of screening the UGC on SM platforms like Facebook, Twitter, Youtube, Instagram with help of large-scale human moderators that make decisions about the appropriateness of UGC (text, image, video) posted on SM (Roberts 2017b). Content moderation is implemented by SM companies in three discrete phases namely (Common 2020).

- Creation: Creation describes the development of the rules (the terms and conditions) that platforms use to govern the user's conduct.
- Enforcement: Enforcement includes flagging problematic content, making decision on whether the content violates the rules set in creation stage and accordingly the action to be taken for the problematic content.
- Response: Response describes the internal appeals process used by platforms and the methods of collective action activists might use to change the platform from the outside. For example, controversies that arose over the live streaming of murder and sexual assaults were considered by social media companies in form of response as announcing hiring more moderators to have better control over such events. (Gibbs 2017).

This section describes the manual, semi-automated and fully-automated methods of moderation.

5.1 Manual approach of moderating detrimental content on SM platforms

Content moderation as defined by Grimmelmann (2015), is the use of administrators or moderators with authority to remove content or prohibit users and making the design decisions that organize how the members of a community

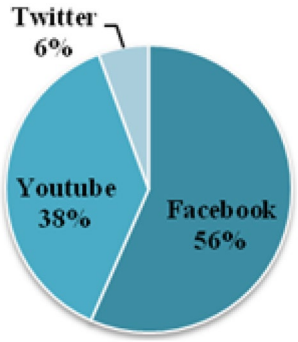


Fig. 14 Manual Content moderation on SM Platforms

engage with one another. Content moderation is considered as indispensable component for SM platforms (Barrett 2020; Roberts 2016). Content moderators are important stakeholders that ensure safety of SM platforms (Roberts 2017; Gillespie 2018; Barrett 2020). The content moderators decide which content is appropriate to be kept on SM and which content should be removed (Barrett 2020).

Commercial content moderation is particularly meant for moderating the objectionable content on SM platforms with help of human moderators that adjudicate such content (Roberts 2016).

The origin of content moderation started with an intention to protect the users of SM platforms from pornography and offensive content (Barrett 2020). Content moderation initially was done by in-house team of people who review the content based on the set of moderation rules defined by social media company and instructions about the removal of certain content (Barrett 2020, Crawford and Gillespie 2016). With the increase in the usage of users and the content shared by them, it became challenging for in-house team to moderate the content. Figure 14 shows the statistics of moderators hired by popular SM platforms (Barrett 2020). As shown in Fig. 14, Facebook holds the highest number of moderators around 15,000 followed by YouTube with 10,000 moderators and Twitter having around 1500 moderators (Barrett 2020). The figures quantify the amount of content

shared on these platforms and the number of moderators who do the task of screening the content. To scale up with the increasing content, social media companies have marginalized the people and have outsourced the task of moderation to third-party vendors who work at different geographical locations which include U.S., Philippines, India, Ireland, Portugal, Spain, Germany, Latvia, and Kenya (Barrett 2020). The task of moderation is also done using online websites like Amazon Mechanical Turk (Roberts 2016).

Flagging is a detection mechanism used by the user community to report an offensive content, violent graphic content to the SM platforms (Gillespie 2018; Roberts 2016; Crawford and Gillespie 2016). To scale with the content published on SM, AI based methods are deployed to detect the detrimental content (Barrett 2020; Crawford and Gillespie 2016). The flagging mechanism is widely observed in the SM platforms that allow the users to express their concern about the content posted on these platforms (Gillespie 2018; Crawford and Gillespie 2016). The flagged content is then by reviewed by the content moderators who checks whether the content violates the Community guideline policies of the platform (Gillespie 2018). Many SM platforms consider the content flagged by the user as important, as it helps in maintaining their brand (Gillespie 2018). The flagging mechanism also reduces the load of content moderators as they need to review only the flagged content instead of reviewing all the posts.

Human content moderators analyze the online comments and posts shared by the users using the Community Guidelines defined by the SM platforms (Roberts 2016). The Community Guidelines are framed by all social media platforms that define the rules and policies about the types of content to be kept and content to be removed from on the platform. For example, Youtube's Community Guidelines include excluding shocking and disgusting content and content featuring dangerous and illegal acts of violence against children (Roberts 2016). Facebook defines Community Standards that include policies on hate speech, targeted violence, bullying, and porn, as well as rules against spam, "false news," and copyright infringement with policy rules made by lawyers, public relations professionals, ex-public policy wonks, and crisis management (Koebler and Cox 2018).

The process of content moderation starts with training of the volunteers about the policies set by the platforms and making them observe the moderation work done by the experts. The volunteers are given the information through the database regarding what constitutes hate speech, violent graphic content (Koebler and Cox 2018) and also includes on-boarding, hands-on practice, and ongoing support and training (Barrett 2020). The moderators are given the task to moderate any specific form of objectionable content. The moderators then decide whether the content is according to the policy standards as defined by the platforms (Barrett 2020). Each moderator is given a handling time to process the content and then make a decision, which is approximately 10–30 s per content Common 2020; Barrett 2020). The moderators after screening the content, remove it, retain it or mark it as disturbing Common 2020; Barrett 2020). SM platforms expect 100% accuracy from content moderators[10] but as Mark Zuckerberg admitted in a white paper that moderators "make the wrong call in more than one out of every 10 cases,"[10].

Moderators also review the content in a different language by using the social media company's proprietary translation software (Barrett 2020). Many times, the moderators had to remove the same content multiple times which have led to many health problems (Barrett 2020; Roberts 2016). Over exposure to disturbing videos and images of sexual assault and violent graphics, the moderators experienced insomnia and nightmares, unwanted memories of troubling images, anxiety, depression, and emotional detachment and suffered from post-traumatic stress disorder (PTSD) (Ofcom 2019; Barrett 2020).

Human experts are involved in pre-moderation phase (moderate the content before it is published) and post-moderation phase (moderate the content after it is published) (Ofcom 2019). The manual approach of moderation requires that the expert must be aware of the context in terms of geographical location and its laws from where the content is shared and published, the SM platform and must be well versed with the language of the content to understand the meaning and the relevance (Roberts 2017a). All these aspects demand a special training for moderators to screen the online content.

5.2 Semi-automated technique of moderating detrimental content on SM platforms

The manual approach of content moderation has many challenges in terms of the volume, veracity and speed of problematic content to be analyzed, the cultural, historical and geographical context around the content. Many companies and governments are proposing automated processes to assist in detection and analysis of problematic content, including disinformation, hate speech, and terrorist propaganda (Leerssen et al. 2020).

Semi-automated moderation techniques include use of AI tools to automatically flag the text, image, video content and review of the flagged content done by the human moderators. The automated flagging mechanism will reduce the workload of human reviewers. The AI based tools like hash matching in which a fingerprint of an image is compared with a database of known harmful images,

[10] https://www.forbes.com/ Facebook Makes 300,000 Content Moderation Mistakes Every Day.

and 'keyword filtering' in which words that indicate potentially harmful content are used to flag content (Ofcom 2019) facilitate the review process of human moderation. The Azure content moderator by Microsoft is AI based content moderation tool that scans text, image, and videos and applies content flags automatically. The web-based Review tool stores and display content for human moderators to assess the content.[11] The tool includes moderation Application Programming Interface (API) that checks the objectionable content like offensive content, sexually explicit or suggestive content, and profanity, checks the images and videos that contain adult or racy content. The review tool assigns or escalates content reviews to multiple review teams, organized by content category or experience level[11].

Andersen et al. (2021) presented a real time moderation of online forums with a Human-In-the-Loop (HiL) to increase the moderation accuracy by exploiting human moderation of uncertain instances in test data. Each comment is classified as valid or blocked using a ML algorithm with an additional comment marked as uncertain which is evaluated and labeled by human moderators. The human labeled instances are added to the training data and then the ML model is re-trained. With moderating 25% of test dataset, the detection of valid comments is increased to 92.30% with help of manual intervention.

The performance of semi-automated techniques of content moderation is more dependent on the accuracy of AI tools used to flag a content and image. The AI tools should also detect the degree of diversity used in the social media UGC which is challenging and demands more attention in the research. The automatic flagging mechanism needs to be experimented in real time and monitor how these tools assist the human moderation process. AI based flagging tools should be exploited more to detect a harmful text or image and give an indication in form of a flag that signifies a terrifying or dreadful content to be screened by a human moderator.

5.3 Automated technique of moderating detrimental content on SM platforms

The psychological trauma experienced by the human moderators (Roberts 2016) and the challenge of handling the significant rise in the UGC on SM platforms demands for a use of automated technologies in the form of AI. With the increasing pressures of government on SM companies to grapple with the disturbing content, both government organization and SM companies are suggesting the use of technical solutions for moderating the SM content (Gorwa et al. 2020). AI and automated systems can assist manual moderation by reducing the amount of content to be reviewed thus increasing the productivity of moderation and also help in restricting the exposure to disturbing content by manual moderators (Ofcom 2019). History reports the use of automated systems like "Automated Retroactive Minimal Moderation" systems to filter the growing spam content on USENET using automated filters (Gorwa et al. 2020).

Systems like automated 'bot' moderators fought vandalism and moderated the articles on Wikipedia (Gorwa et al. 2020). Automated content moderation also referred as algorithmic moderation or algorithmic commercial content moderation are systems that identify, match, predict or classify the UGC which takes the form of text, audio, video or image based on the exact properties and general features of UGC with a decision and governance outcome in form of deletion, blocking the user or removal of account of user (Ofcom 2019; Grimmelmann 2015).Artificial intelligence (AI) is often proposed as an important tool for identifying and filtering out UGC that is offensive or detrimental.

Automated tools are used by the SM platforms to monitor the UGC which covers terrorism content, graphic violence, toxic speech like hate speech and cyberbullying, sexual content, child abuse and spam/fake account detection (Grimmelmann 2015). The Global Internet Forum to Counter Terrorism (GIFCT) is founded by SM platforms like Facebook, Twitter, Microsoft and Youtube to remove the extremists and terrorism content from SM (Ganesh and Jonathan 2020; Grimmelmann 2015). The SM platforms under GIFCT have created a secret database of digital fingerprints (called as 'hash') of terrorist content (images, text, audio, video) called as Shared Industry Hash Database (SIHD) which contain 40,000 image and video hashes (Singh2019) and developed automated systems to detect the terrorist content (Gorwa et al. 2020). The database is updated by adding content through trusted platforms (Grimmelmann 2015). The image or video content uploaded by social media platform users are hashed and checked against the SIHD and if the content matched with hash in database, it is blocked (Gorwa et al. 2020).

Many SM platforms relied on automated techniques of content moderation during COVID 19 pandemic as many human moderators were sent home to limit the exposure to virus (Barrett 2020). Table 18 depicts the automated tools used by SM platforms to moderate the detrimental UGC. The automated tools used by SM platforms make use of ML algorithms that are applied to diverse categories of UGC like text, image video and audio formats. As shown in Table 18, automated tools developed by Facebook like RoBERT architecture detect hate speech in multiple languages

[11] https://docs.microsoft.com/azure/ Content Moderator Overview.

Table 18 Automated Tools to moderate UGC

SM platform	Automated tool	Type of content moderated	Methodology
Google Jigsaw's (Hosseini et al. 2017)	Perspective API	Toxic comments	ML model to score the toxicity of input comments in real time
Microsoft[a]	PhotoDNA	Child exploitation images	Unique digital signature ('hash') for an illegal image compared against a database of another digital signature
Youtube[b]	Content ID	Audio and video	Music and video files uploaded on Youtube are scanned against a database of files
Twitter[c]	Quality Filter	Harassment text	Tool to hide the low-quality notifications from bots and spammers
Counter Extremism Project[d]	eGLYPH	extremist content	Like PhotoDNA, a hash for extremist content
Facebook[e]	RoBERT, RIO, LinFormer	Hate speech content	NLP models in different languages using Transfer learning

RIO Reinforced Integrity Optimizer, *LinFomer* Linear Transformer

[a]https://www.microsoft.com/ Photodna

[b]https://support.google.com/youtube/ YouTube Operations Guide Using Content ID

[c]https://techcrunch.com/2016/08/18/ Twitter is introducing a quality filter to clean up your notifications tab

[d]https://www.counterextremism.com/video/how-ceps-eglyph-technology-works

[e]https://ai.facebook.com/blog/how-ai-is-getting-better-at-detecting-hate-speech, Nov 2020

across Facebook and Instagram.[12]Facebook reported that AI tools like RIO were able to detect 94.7% of hate speech and was removed from Facebook[12]. Tools like PhotoDNA[13] and ContentID[14] work by generating a digital fingerprint called as 'hash' for each of illegal image file or audio and video file. These signatures are stored in a database which is used to compare with other signatures. Signatures identical to stored ones are automatically flagged.[15] As reported by Microsoft[13], PhotoDNA is not face recognition software and hash is not reversible so the tool cannot be used to recreate an image. ML algorithms are used by automated tools for matching the content against the stored database of content which worked best for detection of illegal image.

However, the automated tool named eGLPHY[16] to detect the extremist content raised major concerns about what constitutes an extremist content to be included in the hash database and each platform framed its own policies and definitions of extremist content (Gorwa et al. 2020). This implies a biased decision making as extremist content is subjective and more dependent on geographical location.

6 Discussion and conclusion

SM has brought a big revolution in the society exploring new dimensions of communication through connectivity with people across the globe and providing ample opportunities in professional domain through social media marketing. While SM is proving to be a boom and a kind of blessing to entire society, it is actually a blessing in disguise due to its negative impact which is up surging now with millions of posts on hate speech, online abusive and cyberbullying content, and hundreds of fake news generated by users. Such incidences have led to many fatal deaths, psychological disorders, and depression. This catastrophic negative impact of social media on the society necessitates the dire need of detrimental content detection and moderation. Content detection and moderation is now an inevitable component of SM platforms that is flourishing in real time. This research presents an exhaustive survey with pointers, findings and research gaps involved in detrimental content detection and moderation on social media platforms.

With a phenomenal increase in detrimental content on social media platforms, an accurate detection of such content is important at its first place. Manual detection methods cannot scale up with the increasing detrimental content. The recent advancements in AI through state-of-art algorithms, computational power and the ability to handle huge data (Ofcom 2019) have opened doors to automate the detection process of online content. NLP techniques have shown significant results in parsing the specific form of social media content. Feature engineering techniques like BoW, n-grams,

[12] https://ai.facebook.com/blog/how-ai-is-getting-better-at-detecting-hate-speech, Nov 2020.

[13] https://www.microsoft.com/ Photodna.

[14] https://support.google.com/youtube/ YouTube Operations Guide Using Content ID.

[15] www.snopes.com.

[16] https://www.counterextremism.com/video/how-ceps-eglyph-technology-works.

TF-IDF, and PoS tagging are vital components of NLP that extract the character and word level features from the content and create numerical feature vectors. These frequency-based features representation methods suffer from higher dimensionality and sparse feature vectors which are addressed by word embeddings feature representation techniques. The NLP based ML algorithms perform best when they are trained on a dataset that consists of particular type of content like hate words, abusive words or rumor statements achieving accuracy of around 80% for a specific dataset. In case of hate speech content, there is a spectrum of variation in such content that is dependent on demographic locations, cultures, age, gender, religion. Research have reported the use of ML algorithms for detection of a particular type of hate speech. These algorithms show high rate of false positives when applied to different type of hate speech content. These classifiers lack in ability to capture the nuances in the language used by social media users which needs to be considered in research.

Non-contextual word embeddings like word2vec, GloVe, and contextual word embeddings like FastText, BERT, GPT-3 XLNET, DistilBERT are neural network based pre-trained language models that consider the semantic, syntactic, multilingual, morphological features and Out Of Vocabulary (OOV) words in the text. The maximum accuracy achieved with pre-trained word embeddings alone is around 80%-85%. When using pre-trained models, the number of hyper parameters raise up to billions. Automated systems deploying pre-trained models with these huge parameters leads to increased training time of the neural network model, more compute intensive work which in turn will affect the speed of system. Also, the pre-trained language models trained on huge, uncurated static datasets collected from the Web encode hegemonic views that are harmful to marginalized populations.The pre-trained language models also reveal various kind of bias with more negative sentiments toward specific groups and overrepresentation for words like extremists, toxic and violent content (Bender et al. 2021) This kind of bias in the training characteristics of language models can show a potential risk of wrong judgment when deployed for practical implementation of detection of detrimental content on SM. Transfer learning techniques that make use of pre-trained models is preferred method for detection of English-only content. Automated Systems deploying transfer learning approach have reported low recall due to the variability in definition of a particular content. For example, offensive words misclassified as non-hate words. Although the contextual pre-trained models consider the context of the word in the sentence, the context in social media posts is not considered by these models and the cases of false positives and false negatives is not taken into account by the systems deploying pre-trained models. Exhaustive experimentation and validation are needed before these models are practically deployed.

The present automated systems are dependent on datasets which are created by annotators which has a potential risk of biased decision by the annotator in assigning a label to content. One should also consider the process of automating the annotation which will actually add true essence to the complete automation process. If the current systems focus on manual annotation in developing of automating systems will still end up in designing of semi-automated systems. From the perspective of a fully automated system, automation of this process is also important which is not considered in the present available system Exhaustive research for annotation considering the labeling of data from the angle of the context needs to be operated in these systems.

Traditional ML algorithms need human intervention to extract the important features for detection of inappropriate content. Hand-crafted features are often either over-specified or incomplete. Considering the size of the SM data, developing hand-crafted features for such task is costly and complex job. A bias introduced in developing the features may cause harm in making an incorrect decision by a ML algorithm which restricts its practical deployability in real time. Automatic extraction of features and ability to process huge data by DL techniques through various language models has shown significant results in the task of content moderation. However, DL techniques have difficulty in finding the optimal hyper parameters for a particular dataset (Nasir et al.2021) which increases the training time and inference time during testing. DL techniques also rely on language models which are trained on billions of hyper parameters [for example T-NLG by Microsoft trained on 17 billion parameters (Bender et al. 2021)]. DL techniques along with language models perform sophisticated task but at a cost of increased computational resources and cost, increased training time, and inference time. These aspects restrict their practical implementation in real time. Optimization of DL techniques and fine tuning the language models with optimal parameters is of supreme importance that needs further research.

The present automated system that deploys ML and deep neural networks for detection and classification of detrimental content have considered accuracy, precision and recall as performance metrics. None of the systems to the best of our knowledge have reported the time taken by an algorithm to detect an objectionable content. NLP and neural network models show increased accuracy when they are trained to detect a particular type of detrimental content like abusive speech. These models show decreased accuracy when they applied across different detrimental content format, language and context. Considering practical deployment of these algorithms in real time, time is an inevitable parameter in automated systems. Further research with

rigorous experimentation on the time required will be an important contribution in this domain and therefore needs to be considered.

Content moderation is a process of making an indispensable decision about which form of UGC should be kept online and which form should be removed from the SM platforms. The task of moderation done by SM platforms involves use of human experts that analyze violent, sexually explicit, child abuse content, toxic, illegal hate speech and offensive content in text, image and video format. The experts then flag the content and remove it from the platform if it violates the community guidelines as defined by social media companies. According to statista.com, SM companies spend around $1,440 to $ 28,800 annually on these moderators to review billions of posts every day. With an extensive training of three to four weeks, the moderators evaluate each content within an average time frame of 30 s to 60 s; covering almost 700 posts in an eight-hour shift (Barrett 2020) with accuracy of moderation ranging from 80 to 90%. Considering the time taken by moderators to evaluate content and the accuracy achieved within this stringent time frame is uncertain. The time at which content are moderated is nowhere comparable to the frequency at which posts are published. There are multiple factors involved in manual moderation like training time, the noisy form of content, the mental state of moderators after checking the huge volume of content, understanding the dynamic and reactive community guidelines set by SM platforms before moderation, the amount paid to the moderators, and accuracy of moderation is also not comparable. All these aspects of manual moderation can be bridged by an automated system. Manual moderation also includes reviewing a content that is flagged by a user community on social media. This helps the human moderator but there are more chances of bias getting introduced in the decision of flagged content made by the user community. The context around a content is vital and a crucial aspect which is completely missing in the present manual moderation process.

Semi-automated moderation systems try to deal with the trade-off of volume of content versus the time taken to analyze a content using manual moderation technique. Semi-automated systems are deployed by SM companies to curb with the accelerating increase in the problematic or objectionable content. These systems make use of AI tools to automatically flag a content which is then reviewed by a human moderator. Such systems facilitate the review process of human moderation in evaluating only the objectionable content. However, a particular content flagged by AI tool might not be objectionable from moderator's perspective. This additionally entails a bias and discrepancy in the decision made by the AI tool and the moderator. Transparency in the decision made by AI tool to assist manual moderation is missing and demands immediate attention and further research.

In some typical alarming situations government raises red flags and demands urgent content moderation from social media companies. In such situations, it is challenging for SM companies to appoint manual experts for flagging the flooding content. More ever it has to be done in stipulated time on urgent basis and needs to be done accurately. In such scenarios, automated systems will play a vital role and will obviously be preferred over any semi-automated and manual system. So further research in developing an automated system is a dire need considering such real time situations.

The scalability problem of social media content and psychological trauma experienced by human moderators can be addressed by an automated approach of moderation. The automated approach of content moderation fueled by AI and ML is deployed by many SM platforms in form of automated tools like PhotoDNA by Microsoft, ContentID by YouTube, Quality Filter by Twitter, and RoBERT by Facebook. These tools organize, filter and curate extremists and violent content, child abuse material, hate speech content, and copyright violations in text, image audio and video format. These tools work by creating a common database of illegal images and text content and this database is used by companies to moderate the content. The database is updated with new text and image content. However, each social media platform has their own definition of illegal or harmful text which is stored in the database. This leads to discrepancy in moderating a specific form of content with the possibility of automated tool making an incorrect decision. The definition of an extremist content or hate speech is dependent on the demographic location which is not considered in the current systems. Considering these variations and subjectiveness, it is very important to design and develop an automated system which can be globally deployed across any demographic location and still give encouraging results for content moderation. This is an aspect of paramount importance but has received major attention in the current systems. Therefore, further research is necessary to design globally deployable systems with objectified decision making.

The current trend shows that the user on social media has shown an inclination toward audio and video clips, emojis, smiley's, GIFs formats for expressing their views. The current social media is acutely inclined toward use of these formats. This makes the task of manual moderation too challenging in terms of interpretation, time to evaluate and making a decision about flagging and removing such form of content which can lead to error and affect the accuracy of moderation. An automated approach can assist in moderating multimedia content. However, the current designed automated systems are all focused on words and driven by the content in terms of the text. These systems need exhaustive research to imbibe smileys, emojis, and gifs format so as to make it full proof. To the best of the knowledge, this aspect is completely ignored in the present automated

system. Designing a system that will take into account this multimedia and give the decision is of dire need considering the present scenario. Advanced ML algorithms will be needed to design such systems which are yet not explored. Heavy experimentation and designing the datasets which will include all characteristics of a content making it publicly available, keeping it open for the research community to float their ideas and developing a system that will be universally acceptable is an important aspect that needs to be covered in research.

Google has launched a text translator for 109 languages. Considering a typical case like India, users write regional language in English. Another characteristic of present social media is lack of restricting to one particular language or preference of using combination of languages when expressing and sharing views, (For example writing Marathi in English) called as code mixed language. The liberty of using Hinglish language (Hindi + English) or Reglish (Regional + English) language is another dimension of content moderation that has received little attention in research. Research community has reported creation of datasets in code mixed language for hate speech and abusive content. Multilingual BERT (mBERT) pre-trained models developed by Google which include more than 100 languages are trained on certain code-mixed content (Hinglish) but often lack in detection of fine-grained definition of hate speech content.

Even though the deep neural networks-based NLP models have currently shown promising performance in machine translation, named entity recognition, sentiment analysis, but have underperformed for automated analysis of social media content. It is very important to develop these models that will capture the subtleties of language across different context which needs to be explored in research.

Fairness and trust in decision making by an AI based systems is an important aspect for realization of real time applications. NLP techniques are considered as white-box models that are inherently explainable (Bender et al. 2021). However, due to word embeddings, the present NLP models are based on deep neural network are considered as black box which lack in interpretability. Explainable AI (XAI) (Danilevsky et al. 2020) is a new emerging field of AI aimed at developing a model more explainable and interpretable in terms of making a user understand of how a model arrived at a result. Research literatures have reported various forms of explanation in NLP through feature importance, surrogate model, example driven, provenance, declarative induction (Danilevsky et al. 2020). The explainable aspect is explored for fake news detection (Shu et al. 2019) through attention-based models. XAI though not a fully developed field needs to be explored for developing a transparent automated SM content moderation system with more exploitation on the features extracted from the user's posts on SM.

Further research is needed so as to make context driven decision making about the content is of paramount importance considering manual approach. Content moderation is subjective, and perspective of objectionable language varies according to user, geographic location, culture and history. This all necessitates a exhaustive research and a thorough understanding of social media content while designing a fully automated content moderation system.

The detrimental content posted on the social media has already caused the damage to the society. Present systems focus on moderating it or removing it after the damage has already done. But to the best interest of mankind and humanity, researchers need to think beyond moderating the content and going step further to prevent it wherein there is some flagging assigned to a user and after a threshold is decided on number of inappropriate posts, like ATM cards the user for 24 h is banned from SM. So, the researchers need to think beyond the obvious of only moderating or only restricting their research to moderation of content, but prevention of such cases will actually serve as a boom to social media. Designing a system that will monitor the user's history of posting detrimental content, setting a threshold on the number of objectionable posts and then raising a flag when the threshold has crossed will ensure a safe environment on social media.

References

Ahmed H, Traore I, Saad S (2017) Detection of online fake news using N-Gram analysis and machine learning techniques. In: Traore I, Woungang I, Awad A (eds) Intelligent, secure, and dependable systems in distributed and cloud environments. ISDDC. Lecture Notes in Computer Science, Vol. 10618, pp 127–138. https://doi.org/10.1007/978-3-319-69155-8_9.

Amrutha BR, Bindu KR (2019) Detecting hate speech in tweets using different deep neural network architectures. In: Proceedings of the international conference on intelligent computing and control systems (ICICCS 2019) IEEE, pp 923–926. https://doi.org/10.1109/ICCS45141.2019.9065763.

Andersen JS, Zukunft O, Maalej W (2021) REM: efficient semi-automated real-time moderation of online forums. In: Proceedings of the joint conference of the 59th annual meeting of the association for computational linguistics and the 11th international joint conference on natural language processing: system demonstrations. pp 142–149.

Ayo FE, Folorunso O, Ibharalu FT, Osinuga IA (2020) Machine learning techniques for hate speech classification of twitter data: State-of-the-art, future challenges and research directions. Comput Sci Rev Elsevier. https://doi.org/10.1016/j.cosrev.2020.100311

Badjatiya P, Gupta S, Gupta M, Varma V (2017) Deep learning for hate speech detection in tweets. In: 26th international conference on world wide web companion, Perth, Australia, pp 759–760. https://doi.org/10.1145/3041021.3054223.

Barrett PM (2020) Who moderates the social media giants? A call to end outsourcing. report: NYU Stern Center Centre for Business and Human Rights.

Basile V, Bosco C, Fersini E, Nozza D, Patti V, Pardo FMR, Rosso P, Sanguinetti M (2019) Semeval-2019 task 5: Multilingual

detection of hate speech against immigrants and women in twitter. In: Proceedings of the 13th international workshop on semantic evaluation, pp 54–63. https://doi.org/10.18653/v1/S19-2007.

Bender EM, Gebru T, Shmitchell S, McMillan A (2021) On the dangers of stochastic parrots: can language models be too big? In: Proceedings of the 2021 ACM conference on fairness, accountability, and transparency, pp 610–623. https://doi.org/10.1145/3442188.3445922.

Boididou C, Papadopoulos S, Nguyen DT, Boato G, Kompatsiaris Y(2015) The certh-unitn participation@ verifying multimedia use 2015. Verifying multimedia use at MediaEval 2015. In: MediaEval benchmarking initiative for multimedia evaluation.

Bojanowski P, Grave E, Joulin A, Mikolov T (2017) Enriching word vectors with subword information. Trans Assoc Comput Linguistics 5:135–146. https://doi.org/10.1162/tacl_a_00051

Bonet OG, Miguel NP, Garcia-Pablos A, Cuadros M (2018) Hate speech dataset from a white supremacy forum. In: 2nd workshop on abusive language online @ EMNLP. https://doi.org/10.18653/v1/W18-5102.

Brown TB et al (2020). Language models are few-shot learners. arXiv:2005.14165v4 [cs.CL]

Burfoot C, BaldwinT (2009) Automatic satire detection: are you having a laugh? In: Proceedings of the ACL-IJCNLP 2009 conference short papers, pp 161–164.

Burnap P, Williams ML (2015) Cyber hate speech on twitter: an application of machine classification and statistical modeling for policy and decision making. Policy Internet 7(2):223–42. https://doi.org/10.1002/poi3.85

Cao J, Qi P, Sheng Q, Yang T, Guo J, Li J (2020) Exploring the role of visual content in fake news detection. arXiv:2003.05096v1 [cs.MM].

Chakraborty A, Paranjape B, Kakarla S, Ganguly N (2016) Stop clickbait: Detecting and preventing clickbaits in online news media. In: IEEE/ACM international conference on advances in social networks analysis and mining, pp 9–16. https://doi.org/10.1109/ASONAM.2016.7752207.

Cheng L, Li J, Silva Y, Hall D, Liu H (2019) Xbully: cyberbullying detection within a multi-modal context. In: Proceedings of the twelfth ACM international conference on web search and data mining, pp 339–347. https://doi.org/10.1145/3289600.3291037.

Chung YL, Kuzmenko E, Tekiroglu SS, Guerini M (2019) CONAN–COunter NArratives through Nichesourcing: a multilingual dataset of responses to fight online hate speech. In: Proceedings of the 57th annual meeting of the association for computational linguistics, pp 2819–2829. https://doi.org/10.18653/v1/P19-1271.

Colomina C, Margalef HS, Youngs R (2021) The impact of disinformation on democratic processes and human rights in the world. Policy Depart Director-General External Policies. https://doi.org/10.2861/677679

Common MF (2020) Fear the Reaper: how content moderation rules are enforced on social media. Int Rev Law Comput Technol. https://doi.org/10.1080/13600869.2020.1733762

Crawford K, Gillespie T (2016) What is a flag for? social media reporting tools and the vocabulary of complaint. New Media Soc 18(3):410–428. https://doi.org/10.1177/1461444814543163

Cui L, Lee D (2020) CoAID: COVID-19 healthcare misinformation dataset. arXiv:2006.00885

Cui L, Wang S, Lee D (2019) SAME: sentiment-aware multi-modal embedding for detecting fake news. IEEE/ACM international conference on advances in social networks analysis and mining (ASONAM), pp 41–48. https://doi.org/10.1145/3341161.3342894.

Cunha et al (2021) On the cost-effectiveness of neural and non-neural approaches and representations for text classification: A comprehensive comparative study. Inf Process Manag 58(3):102481

Danilevsky M, Qian K, Aharonov R, Katasis Y, Kawas B, Sen P (2020) A Survey of the state of explainable AI for natural language processing. arXiv: 2010.00711v1 [cs.CL].

Davidson T, Warmsley D, Macy M, Weber I (2017) Automated hate speech detection and the problem of offensive language. In: Proceedings of the 11th international AAAI social media, ICWSM ', vol 17, 512–515.

Devlin J, Chang M, Lee K, Toutanova K (2019). BERT: pre-training of deep bidirectional transformers for language understanding. arXiv:1810.04805.

Dinakar K, Reichart R, Lieberman H (2011) Modeling the detection of textual cyberbullying. In: Fifth international AAAI conference on weblogs and social media, pp 11–17.

Duarte N, Llanso E, Loup A (2017) Mixed Messages? The limits of automated social media content analysis.

Elhadad MK, Li KF, Gebali F (2020) A novel approach for selecting hybrid features from online news textual metadata for fake news detection. 3PGCIC 2019. LNNS 96:914–925. https://doi.org/10.1007/978-3-030-33509-0_86

Ellison NB (2007) Social network sites: Definition, history, and scholarship. J Computer-Mediated Commun 13(1):210–230

Fortuna P, Nunes S (2018) A survey on automatic detection of hate speech in text. ACM Comput Surv 51(4):1–30. https://doi.org/10.1145/3232676

Gambäck B, Sikdar UK (2017) Using convolutional neural networks to classify hate-speech. In: Proceedings of the first workshop on abusive language online, pp 85–90. https://doi.org/10.18653/v1/W17-3013.

Ganesh B, Jonathan B (2020) Countering extremists on social media: challenges for strategic communication and content moderation. Policy Internet 2(1):6–19. https://doi.org/10.1002/poi3.236

Gibbs S (2017) Facebook live: Zuckerberg adds 3000 moderators in wake of murders. https://www.theguardian.com/technology/2017/may/03/facebook-live-zuckerberg-adds-3000-moderators-murders. Accessed 17 October 2021.

Gillespie T (2018) Custodians of the internet: platforms, content moderation, and the hidden decisions that shape social media. Yale University Press, New Haven

Gitari ND, Zuping Z, Damien H, Long J(2015) A Lexicon-based Approach for Hate Speech Detection. In: International Journal of Multimedia and Ubiquitous Engineering Vol.10 (4): 215-230. https://doi.org/10.14257/ijmue.2015.10.4.21

Glazkova, A., Glazkov, M., Trifonov, T. (2021). g2tmn at Constraint@AAAI2021: Exploiting CT-BERT and Ensembling Learning for COVID-19 Fake News Detection. In: Chakraborty, T., Shu, K., Bernard, H.R., Liu, H., Akhtar, M.S. (eds) combating online hostile posts in regional languages during emergency situation. CONSTRAINT 2021. Communications in Computer and Information Science, vol 1402. Springer, Cham. doi: https://doi.org/10.1007/978-3-030-73696-5_12

Golbeck et al (2017) A large, labeled corpus for online harassment research. In: WebSci '17: proceedings of the 2017 ACM on web science conference, 229–233. https://doi.org/10.1145/3091478.3091509.

Goldani MH, Momtazi S, Safabakhsh R (2020a) Detecting fake news with capsule neural networks. Appl Soft Comput J. https://doi.org/10.1016/j.asoc.2020.106991

Goldani MH, Safabakhsh R, Momtazi S (2020b) Convolutional neural network with margin loss for fake news detection. Inf Process Manage 58:1–12. https://doi.org/10.1016/j.ipm.2020.102418

Gorwa R, Binns R, Katzenbach C (2020) Algorithmic content moderation: technical and political challenges in the automation of platform governance. Big Data Soc. https://doi.org/10.1177/2053951719897945

Granik M, Mesyura V (2017) Fake news detection using naive Bayes classifier. In: IEEE first Ukraine conference on electrical and computer engineering (UKRCON), pp 900–903. https://doi.org/10.1109/UKRCON.2017.8100379.

Grimmelmann J (2015) The virtues of moderation. Yale J Law Technol. https://doi.org/10.31228/osf.io/qwxf5

Hamdi T, Slimi H, Bounhas I, Slimani Y (2020) A hybrid approach for fake news detection in twitter based on user features and graph embedding. ICDCIT 2020. LNCS 11969:266–280. https://doi.org/10.1007/978-3-030-36987-3_17

Hirschberg J, Manning HD (2015) Advances in natural language processing. Science 349(6245):261–266. https://doi.org/10.1126/science.aaa8685

Horne BD, Adali S (2017) This just in: fake news packs a lot in title, uses simpler, repetitive content in text body, more similar to satire than real news. In: The workshops of the eleventh international AAAI conference on web and social media AAAI technical report WS-17, News and Public Opinion, pp 759–766.

Hosseini H, Kannan S, Zhang B,Poovendran R (2017) Deceiving Google's perspective API built for detecting toxic comments. arXiv:1702.08138v1 [cs.LG].

Islam MdR, Liu S, Wang X, Xu G (2020) Deep learning for misinformation detection on online social networks: a survey and new perspectives. Social Netw Anal Mining. https://doi.org/10.1007/s13278-020-00696-x

Kaplan AM, Haenlein M (2010) Users of the world, unite! The challenges and opportunities of Social Media. Business Horizons 53(1):59–68

Khattar D, Goud JS, Gupta M, Varma V(2019) MVAE: multimodal variational autoencoder for fake news detection. In: The World Wide Web conference (WWW '19). Association for computing machinery, New York, NY, USA, pp 2915–2921. https://doi.org/10.1145/3308558.3313552

Kocoń J, Figas A, Gruza M, Puchalska D, Kajdanowicz T, Kazienko P (2021) Offensive, aggressive, and hate speech analysis: from data-centric to human-centered approach. Inf Process Manag. https://doi.org/10.1016/j.ipm.2021.102643

Koebler J, Cox J (2018) The impossible job: inside Facebook's struggle to moderate two billion people. https://www.vice.com/en/article/how-facebook-content-moderation-works. Accessed on 25 October 2021.

Kumar G, Singh JP, Kumar A (2021) A Deep Multi-modal neural network for the identification of hate speech from social media. IFIP Int Feder Inf Process LNCS 12896:670–680. https://doi.org/10.1007/978-3-030-85447-8_55

Kumar R, Ojha AK, Malmasi S, Zampieri M (2018) Benchmarking Aggression Identification in Social Media. In: Proceedings of the First Workshop on Trolling, Aggression and Cyberbullying (TRAC-2018):1–11

Kumari K, Singh JP, Dwivedi YK, Rana NP (2021) Multi-modal aggression identification using convolutional neural network and binary particle swarm optimization. Future Gener Comput Syst 118:187–197. https://doi.org/10.1016/j.future.2021.01.014

Kwok I, Wang Y (2013) Locate the hate: detecting tweets against blacks. In: Proceedings of the twenty-seventh AAAI conference on artificial intelligence, AAAI'2013, pp 1621–1622.

Lan Z et al (2020). ALBERT: A LITE BERT for self-supervised learning of language representations. arXiv:1909.11942v6 [cs.CL]

Leerssen P, Hoboken J V, Harambon J, Lanso E (2020) Artificial Intelligence, Content Moderation, and Freedom of Expression, Transatlantic Working Group.

Li L, Levi O, Hosseini P, Broniatowski D (2020). A multi-modal method for satire detection using textual and visual cues: In: Proceedings of the 3rd NLP4IF workshop on NLP for internet freedom: censorship, disinformation, and propaganda, barcelona, Spain (Online). International Committee on Computational Linguistics (ICCL), pp 33–38

Li L, Levi O, Hosseini P, Broniatowski DA (2021). A multi-modal method for satire detection using textual and visual cues. arXiv:2010.06671v1 [cs.CL]

Liu H, Burnap P, Alorainy M, Williams ML (2019) A fuzzy approach to text classification with two-stage training for ambiguous instances. IEEE Trans Comput Soc Syst 6(2):227–240. https://doi.org/10.1109/TCSS.2019.2892037

Liu Y et al (2019). RoBERTa: a robustly optimized BERT pretraining approach. arXiv:1907.11692v1 [cs.CL]

Ma J, Gao W, Wong KF (2019) Detect rumors on twitter by promoting information campaigns with generative adversarial learning. In: Proceedings of the 28th international conference on World Wide Web, ACM: 3049–3055. doi:https://doi.org/10.1145/3308558.3313741.

Malmasi S, Zampieri M (2018) Challenges in discriminating profanity from hate speech. J Exp Theor Artif Intell 30:187–202. https://doi.org/10.1080/0952813X.2017.1409284

Mandl T, Modha S, Majumder P, Patel D, Dave M, Mandlia C, Patel A (2019) Overview of the HASOC track at FIRE 2019: hate speech and offensive content identification in Indo-European languages. In: Proceedings of the 11th forum for information retrieval evaluation (FIRE '19). Association for Computing Machinery, New York, USA, pp 14–17. https://doi.org/10.1145/3368567.3368584.

Mangalam K, Kumar A (2019) Section 66A: an unending saga of misuse and harassment. https://lawschoolpolicyreview.com/2019/06/04/

Mathur P, Shah R, Sawhney R, Mahata D (2018) Detecting offensive tweets in Hindi-English code-switched language. In: Proceedings of the sixth international workshop on natural language processing for social media, pp 18–26. https://doi.org/10.18653/v1/W18-3504.

Mikolov T, Le QV, Sutskever I (2013) Exploiting similarities among languages for machine translation. arXiv:1309.4168v1 [cs.CL].

Mitra T, Gilbert E (2015) Credbank: A largescale social media corpus with associated credibility annotations. Proc Int AAAI Conf Web Social Media 9(1):258–267

Modha S, Majumder P, Mandl T, Mandalia C (2020) Detecting and visualizing hate speech in social media: A cyber Watchdog for surveillance. Expert Syst Appl. https://doi.org/10.1016/j.eswa.2020.113725

Mollas I, Chrysopoulou Z, Karlos S, Tsoumakas G (2021). ETHOS: an online hate speech detection dataset. arXiv:2006.08328v2 [cs.CL].

Mutanga RT, Naicker N, Olugbara OO (2020). Hate speech detection in twitter using transformer methods, pp 614–620. (IJACSA) Int J Adv Comput Sci Appl, 11(9)

Naeem SB, Bhatti R, and Khan A (2021). An exploration of how fake news is taking over social media and putting public health at risk. Health Info Libr J 38(2):143–149. https://doi.org/10.1111/hir.12320. Epub 2020 Jul 12. PMID: 32657000; PMCID: PMC7404621.

Nascimento et al (2022) An overview of systematic reviews of the current state of the art of infodemics and health misinformation and its repercussions in public health: recommendations, challenges, and available research opportunities. Bulletin of the World Health Organization. May 2022.

Naseem U, Razzak I, Hameed IA (2019) Deep context-aware embedding for abusive and hate speech detection on twitter. Australian J Intell Inf Process Syst 15(4):69–76

Nasir JA, Khan OS, Varlamis I (2021) Fake news detection: A hybrid CNN-RNN based deep learning approach. Int J Inf Manag Data Insights 1(1):1–13. https://doi.org/10.1016/j.jjimei.2020.100007

Ngai EWT, Tao SSC, Moon KKL (2015) Social media research: Theories, constructs, and conceptual frameworks. Int J Inf Manage 35(1):33–44. https://doi.org/10.1016/j.ijinfomgt.2014.09.004

Nguyen VH, Sugiyama K, Nakov P, Kan MY(2020) FANG: leveraging social context for fake news detection using graph representation. arXiv:2008.07939v2 [cs.SI].

Nobata C, Tetreault JR, Thomas A, Mehdad Y, Chang Y (2016) Abusive language detection in online user content. In: Proceedings of the 25th international conference on World Wide Web, pp 145–153. https://doi.org/10.1145/2872427.2883062 .

P´erez-Rosas V, Kleinberg B, Lefevre A, Mihalcea R (2017) Automatic detection of fake news. arXiv: 1708.07104.

Paka WS, Bansal R, Kaushik A, Sengupta S, Chakraborty T (2020) Cross-SEAN: A cross-stitch semi-supervised neural attention model for COVID-19 fake news detection. Appl Soft Comput. https://doi.org/10.1016/j.asoc.2021.107393

Pan SJ, Yang Q (2010) A survey on transfer learning. IEEE Trans Knowl Data Eng 22(10):1345–1359. https://doi.org/10.1109/TKDE.2009.191

Papakyriakopoulos O, Medina Serrano JC, Hegelich S (2020) The spread of COVID-19 conspiracy theories on social media and the effect of content moderation. The Harvard Kennedy School (HKS) Misinform Rev. https://doi.org/10.37016/mr-2020-034

Park JH, Fung P (2017) One-step and two-step classification for abusive language detection on twitter. ALW@ACL: 41–45. https://doi.org/10.18653/v1/w17-3006.

Patwa P et al. (2021). Fighting an infodemic: COVID-19 fake news dataset. combating online hostile posts in regional languages during emergency situation.In: CONSTRAINT 2021. Communications in Computer and Information Science, vol 1402. Springer, Cham. https://doi.org/10.1007/978-3-030-73696-5_3.

Pennington G, Socher R, Manning CD (2014) Glove: Global vectors for word representation. In: Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP), pp 1532–1543. doi:https://doi.org/10.3115/v1/D14-1162.

Peters M, Neumann M, Iyyer M, Gardner M, Clark C, Lee K, Zettlemoyer L (2018) Deep contextualized word representations. In Proceedings of the 2018 conference of the north american chapter of the association for computational linguistics: human language technologies, vol 1, pp 2227–2237. https://doi.org/10.18653/v1/N18-1202.

Pitsilis GK, Ramampiaro H, Langseth H (2018) Effective hate-speech detection in Twitter data using recurrent neural networks. Appl Intell 48(12):4730–4742. https://doi.org/10.1007/s10489-018-1242-y

Qi P, Cao J, Yang T, Guo J, Li J (2019) Exploiting multi-domain visual information for fake news detection. In: 2019 IEEE international conference on data mining (ICDM), pp 517–527. https://doi.org/10.1109/ICDM.2019.00062.

Qian F, Gong C, Sharma K, Liu Y (2018) Neural user response generator: fake news detection with collective user intelligence. In: Proceedings of the twenty-seventh international joint conference on artificial intelligence (IJCAI-18), pp 3834–3840. https://doi.org/10.24963/ijcai.2018/533.

Ofcom Report (2019) Use of AI in online content moderation.

Roberts ST (2016) Commercial content moderation: digital laborers' dirty work. In: Media Studies Publications. 12. https://ir.lib.uwo.ca/commpub/12

Roberts ST (2017a) Content moderation. UCLA Previously Published Works, pp 1–6

Roberts ST (2017b) Social media's silent filter. https://www.theatlantic.com/technology/archive/2017b/03/commercial-content-moderation/518796/ Accessed 17 October 2021.

Robinson D, Zhang Z, Tepper J (2018) Hate speech detection on Twitter: feature engineering v.s. feature selection. In: Proceedings of the 15th extended semantic web conference, pp 46–49, 2018. doi: https://doi.org/10.1007/978-3-319-98192-5_9.

Ross B, Rist M, Carbonell G, Cabrera B, Kurowsky N, Wojatzki M(2016) Measuring the reliability of hate speech annotations: the case of the European Refugee Crisis. In: Proceedings of NLP4CMCIII:3rd workshop on natural language processing for computer-mediated communication (Bochum), vol. 17, pp 6–9. https://doi.org/10.17185/duepublico/ 42132.

Roy PK, Tripathy AK, Das TK, Gao XZ (2020) A framework for hate speech detection using deep convolutional neural network. IEEE Access 8:204951–204962. https://doi.org/10.1109/ACCESS.2020.3037073

Rubin VL, Conroy N, Chen Y, Cornwell S (2016) Fake news or truth? Using satirical cues to detect potentially misleading news. In: Proceedings of the second workshop on computational approaches to deception detection, pp 7–17. https://doi.org/10.18653/v1/W16-0802.

Ruchansky N, Seo S, Liu Y(2017) CSI: a hybrid deep model for fake news detection. In: Proceedings of the 2017 ACM on conference on information and knowledge management, ACM, pp 797–806. https://doi.org/10.1145/3132847.3132877.

Ruckenstein M, Turunen LL (2020) Re-humanizing the platform: content moderators and the logic of care. New Media Soc 22(6):1026–1042

Sahu G, Cohen R, Vechtomova O (2021) Towards a multi-agent system for online hate speech detection. arXiv:2105.01129v1 [cs.AI].

Sanh V, Debut L, Chaumond J, and Wolf T (2020). DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. arXiv:1910.01108v4 [cs.CL]

Schmidt A, Wiegand M (2017) A survey on hate speech detection using natural language processing. In: Proceedings of the fifth international workshop on natural language processing for social media, pp 1–10. https://doi.org/10.18653/v1/W17-1101.

Sharma S, Agrawal S, Shrivastava M (2018) Degree based classification of harmful speech using twitter data. arXiv:1806.04197.

Shu K, Sliva A, Wang S, Tang J, Liu H (2017) Fake news detection on social media: a data mining perspective. ACM SIGKDD Explorations Newsl 19(1):22–36. https://doi.org/10.1145/3137597.3137600

Shu K, Mahudeswaran D, Wang S, Lee D, Liu H (2018) Fakenewsnet: A data repository with news content, social context and dynamic information for studying fake news on social media. arXiv:1809.01286v3 [cs.SI].

Shu K, Cui L, Wang S, Lee D, Liu H (2019) DEFEND: explainable fake news detection. In: Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery data mining, pp 395–405. https://doi.org/10.1145/3292500.3330935.

Singh S (2019) Everything in Moderation. https:// newamerica.org/oti/reports/everything-moderation-analysis-how-internet-platforms-are-using-artificialintelligence-moderate-user-generated-content.

Singh JP, Kumar A, Rana N, Dwivedi Y (2020) Attention-based LSTM network for rumor veracity estimation of tweets. Inf Syst Front. https://doi.org/10.1007/s10796-020-10040-5

Singhal S, Shah RR, Chakraborty T, Kumaraguru P, Satoh S (2019) SpotFake: a multi-modal framework for fake news detection. IEEE fifth international conference on multimedia big data (BigMM), pp 39–47. https://doi.org/10.1109/BigMM.2019.00-44.

Singhania S, Fernandez N, Rao S (2017) 3HAN: a deep neural network for fake news detection. ICONIP 2017. Part II, LNCS 10635:1–10. https://doi.org/10.1007/978-3-319-70096-0_59

Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, Kaiser L, Polosukhin I (2017) Attention is all you need. Adv Neural Inf Process Syst 30:5998–6008

Verma G, Srinivasan BV (2019) A lexical, syntactic, and semantic perspective for understanding style in text. arXiv:1909.08349 v1 [cs.CL].

Vigna FD, Cimino A, Dell'Orletta F, Petrocchi M, Tesconi M (2017). Hate me, hate me not: hate speech detection on facebook. In: Proceedings of the first Italian conference on cybersecurity (ITASEC17), 86–95.

Vijayarani S, Ilamathi J, Nithya S (2015) Preprocessing techniques for text mining - an overview. Int J Comput Sci Commun Netw 5(1):7–16

Wang WY (2017) Liar, liar pants on fire: a new benchmark dataset for fake news detection. arXiv:1705.00648v1 [cs.CL] .

Wang B, Ding H (2019). YNU NLP at SemEval-2019 task 5: attention and capsule ensemble for identifying hate speech. In: Proceedings of the 13th international workshop on semantic evaluation (SemEval-2019), pp 529–534

Wang Y, Ma F, Jin Z, Yuan Y, Xun G, Jha K, Su L, Gao J (2018) EANN: event adversarial neural networks for multi-modal fake news detection. In: Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining, pp 849–857. https://doi.org/10.1145/3219819.3219903.

Waseem Z (2016) Are you a racist or am I seeing things? Annotator influence on hate speech detection on Twitter. In: Proceedings of the first workshop on NLP and computational social science, pp 138–142. https://doi.org/10.18653/v1/W16-5618.

Waseem Z, Hovy D (2016) Hateful symbols or hateful people? Predictive features for hate speech detection on twitter, In: Proceedings of NAACL-HLT, pp 88–93.

Watanabe H, Bouazizi M, Ohtsuki T (2018) Hate speech on twitter a pragmatic approach to collect hateful and offensive expressions and perform hate speech detection. IEEE Access 6:13825–13835. https://doi.org/10.1109/ACCESS.2018.2806394

Wendling M (2018) The (almost) complete history of 'fake news'. https://www.bbc.com/news/blogs-trending-42724320. Accessed 12 October 2021.

Wyrwoll C (2014) User-generated content. In: Social media, pp 11–45. https://doi.org/10.1007/978-3-658-06984-1_2.

Yang Z, Wang C, Zhang F, Zhang Y, Zhang H (2015) Emerging rumor identification for social media with hot topic detection. In: 12th web information system and application conference (WISA), pp 53–58. https://doi.org/10.1109/WISA.2015.19.

Yang Z et al (2020). XLNet: generalized autoregressive pretraining for language understanding. arXiv:1906.08237v2 [cs.CL]

Zhang X, Ghorbani AA (2020) An overview of online fake news: characterization, detection, and discussion. Inf Process Manag. https://doi.org/10.1016/j.ipm.2019.03.004

Zhang Z, Luo L (2019) Hate speech detection: a solved problem? The challenging case of long tail on twitter. Semantic Web 1:925–945

Zhang Z, Robinson D, Tepper J (2018) Detecting hate speech on twitter using a convolution-GRU based deep neural network. ESWC 2018 LNCS 10843:745–760. https://doi.org/10.1007/978-3-319-93417-4_48

Zhang Q, Zhang S, Dong J, Xiong J, Cheng X (2015) Automatic detection of rumor on social network. LNAI 9362, NLPCC, pp 113–122. https://doi.org/10.1007/978-3-319-25207-0_10.

Zhang J, Dong B, Yu PS (2019) FAKEDETECTOR: effective fake news detection with deep diffusive neural network. arXiv:1805.08751v2 [cs.SI].

Zhong H, Li H, Squicciarini AC, Rajtmajer SM, Griffin C, Miller DJ Caragea C (2016) Content-driven detection of cyberbullying on the instagram social network. In: IJCAI, proceedings of the twenty-fifth international joint conference on artificial intelligence, pp 3952– 3958

Zhou Y, Yang Y, Liu H, Liu X, and Savage N (2020) Deep learning based fusion approach for hate speech detection, pp 128923–128929.IEEE Access, vol. 8, https://doi.org/10.1109/ACCESS.2020.3009244.

Zhou X, Wu J, Zafarani R (2020b) SAFE: similarity-aware multi-modal fake news detection. In: The 24th pacific-asia conference on knowledge discovery and data mining, LNAI 12085: 354–367, 2020b. https://doi.org/10.1007/978-3-030-47436-2_27.

Zhou X, Zafarani R (2020) A survey of fake news: fundamental theories, detection methods, and opportunities. ACM Comput Surv 53(5):1–30. https://doi.org/10.1145/3395046

Systematic Literature Review: Toxic Comment Classification

2022 IEEE 7th International Conference on Information Technology and Digital Applications (ICITDA) | 978-1-6654-6136-8/22/$31.00 ©2022 IEEE | DOI: 10.1109/ICITDA55840.2022.9971338

Felix Museng
Computer Science Department
School of Computer Science
Bina Nusantara University
Jakarta, Indonesia 11480
felix.museng@binus.ac.id

Adelia Jessica
Computer Science Department
School of Computer Science
Bina Nusantara University
Jakarta, Indonesia 11480
adelia.jessica@binus.ac.id

Nicole Wijaya
Computer Science Department
School of Computer Science
Bina Nusantara University
Jakarta, Indonesia 11480
nicole002@binus.ac.id

Anderies Anderies
Computer Science Department
School of Computer Science
Bina Nusantara University
Jakarta, Indonesia 11480
anderies@binus.ac.id

Irene Anindaputri Iswanto
Computer Science Department
School of Computer Science
Bina Nusantara University
Jakarta, Indonesia 11480
irene.iswanto@binus.ac.id

***Abstract*—Over the last decade, deep learning models have surpassed machine learning models in text classification. However, with the continuity of the digital age, many are exposed to the dangers of the internet. One of the dangers would be cyberbullying. In an attempt to decrease cyberbullying, much toxic text detection and classification research has been done. In this paper, we aim to understand the effectiveness of deep learning models compared to machine learning models along with the most common models used by researchers in the last 5 years. We will also be providing insight on the most common data sets utilized by researchers to detect toxic comments. To achieve this, we have compiled the datasets of research papers and analyze the algorithm used. The findings indicate that Long Term Short Memory is the most routinely mentioned deep learning model with 8 out of 26 research papers. LSTM has also repeatedly yielded high accuracy results with above 79% for around 9000 data which could be adjusted depending on the pre-processing method used. There have been attempts to combine more than one deep learning algorithms, however these hybrid models might not result in a better accuracy than an original model. Furthermore, the most frequent sources of datasets came from Kaggle and Wikipedia datasets and a total of 13 researchers that used Wikipedia's talk page edits as their dataset.**

***Keywords*—Deep Learning, Machine Learning, Toxic Text Classification, LSTM**

I. INTRODUCTION

Toxic is a term commonly used to describe something as unpleasant. Toxic comments are comments made by users that use unpleasant words which could offend other users. These unpleasant words could be rude, disrespectful, and degrading.

With the increasing trend of social media, cyberbullying has quickly become one of the most concerning issues in this modern era. More often than not, a person would most likely have had experience of being cyberbullied or even witness a cyberbullying case. Although there are some unsevere cyberbullying, for example calling someone stupid online, there are also a lot of extreme cases of cyberbullying. Therefore, as a way to prevent cyberbullying, most companies have adopted some kind of toxic comment detection such as blurring or changing the toxic words into other symbols such as *. Currently, there are many methods used to detect toxic comments. Most of them are Machine Learning methods.

Machine learning has become one of the ways to detect toxic comments. Different machine learning algorithms were being used to classify toxic comments i.e., logistic regression, random forest, SVM (Support Vector Machine), Naive Bayes, decision tree, and KNN (K-Nearest Neighbors) [18]. Most researchers have done these algorithms in different datasets, but based on the accuracy score, some models still need more improvements.

Therefore, traditional machine learning methods tend to have a low precision for toxic comments detection as the variation of cyberbullying style is high and mostly focused on the usage of swear words, hence the usage of deep learning methods is used to improve the accuracy for detection of cyberbullying [19].

Deep learning is known as a part or subset of machine learning. Deep learning eliminates some data pre-processing and can distinguish important data that would alter the results. An example of deep learning would be being able to do things that came naturally as a human. A popular representation of deep learning implementation in the real world would be driverless cars. There are a couple of frequently mentioned algorithms in deep learning. Those are CNN (Convolutional Neural Networks) [10], RNN (Recurrent Neural Networks) [2], and LSTM (Long Short-Term Memory Networks) [9].

Deep learning has some advantages compared to machine learning. One of the advantages of deep learning is that it can handle a large amount of data to be processed, especially when it comes to complicated issues like picture classification, natural language processing, and speech recognition. The objective of this paper is to find relevant information from research papers on the toxic comment classification and systematically listing and comparing the existing research which then the information is used to help direct future research.

The research questions and the standards for the paper that we employed are included in section 2. We aim to understand how successful deep learning is in comparison to machine learning techniques. As a result, in section 3, we classify the datasets utilized and the algorithms employed in papers. Finally, we provide our paper's conclusion in section 4.

II. STUDY REVIEW

A. Planning

The first step in the planning process is to identify the requirements for a particular systematic review. In the Introduction section, we discussed the necessity for a comprehensive review of machine learning algorithms for toxic comment classification. Following that, we established the following primary research questions:

- RQ1: Which data sets are utilized to detect toxic comments?
- RQ2: What are the most common deep learning models used to detect toxic comments?

Based on the research questions of the study, we have defined several electronic databases as the source of papers and journals. To ensure the eligibility of the references selected, Selection and Exclusion criteria are applied.

Table I. *Selection Criteria*

Inclusion Criteria	**Exclusion Criteria**
The paper must be related to classifying toxic comments with machine learning.	The work was published before 2018.
The paper must propose a method for filtering the toxic comments.	Non-relative studies to the research questions.
The work is recognized internationally.	-

B. Literature Review

In today's world, everyone has their own social media account where they may easily engage with others over the internet. This, however, can lead to cyberbullying, in which people try to make fun of others or harass, threaten, embarrass, or target someone else. For some people, this online brawl frequently turns into real-life threats, such as suicide [25].

Researchers are utilizing machine learning to detect and classify toxic comments to minimize cyberbullying. There are many models in machine learning to do it, such as Naive Bayes, Support Vector Machine (SVM), Random Forest, Logistic Regression, Decision tree, and K-Nearest Neighbor (KNN) [18].

However, in recent years, researchers prefer to use deep learning algorithms instead of machine learning. In the research [22], it was discovered that the proposed method, which combines CNN+LSTM, performs the best with The Unhealthy Comments Corpus (UCC) dataset available in GitHub which yields an accuracy score of 88.76%. It exceeded logistic regression and SVM, which had accuracy scores of 57.54 percent and 69.15 percent, respectively.

In another research [2], it is found that Multi-Emb RNN-CNN (0.59) outperformed Linear Regression (0.38) in F1 scores, further solidifying that Deep Learning algorithms perform better than Machine Learning algorithms. Taking reliability as a context, LIME model interpretability techniques could be used to analyze each method to help find the best model even if the scores are high [3].

Studies by Almerekhi et al [7], [6] utilize LSTM with GloVe on Reddit datasets to find variables that can help detect toxic comments such as toxic triggers, toxic posting behavior, and moderator rules strictness, the study also found evidence that some community is more robust against toxicity.

A comparative study by Zhao et al. [26] found the usage of language models such as BERT and RoBerta combined with CNN and BiLSTM on 4 different datasets. Another study by Kim et al. [6] utilizes GloVe, Word2Vec, Paragram, and FastText combined with BiLSTM, after implementing the topic-enhanced word embedding, the model is able to achieve an F1 score of 0.667.

C. Overview

The main research papers that we used were published in the range of 2018 until 2022 (within 5 years). The highest number of studies was in 2021. In 2022, there are just a few of them because it's the current year. In addition, all the research papers that we reference consist of fourteen international conference and journal papers, and twelve book chapters.

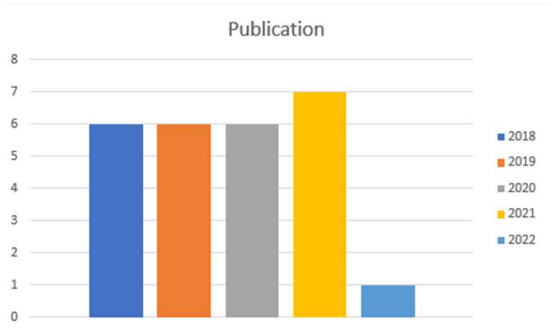


Fig 1. *Publication Year*

For the search criteria, we have given a detailed explanation of it in the table below (Table II).

Table II. *Search Criteria*

Database Searched	Search Terms / Keywords	Limits	Initial Search	Included in paper
IEEE Xplore	"toxic comment classification"	Between 2018 until 2022	35	[9], [11], [12], [18]
	"cyberbullying in social media"	Between 2018 until 2022, Publication topic: "learning", "deep learning"	49	[13], [25]
	"sentiment analysis deep learning"	Between 2018 until 2022,	758	[5]
Science Direct	"classify toxic comments deep learning"	Between 2018 until 2022, computer science subject areas, research articles	69	[1], [22]

(Continued) Table II. *Search Criteria*

Database Searched	Search Terms / Keywords	Limits	Initial Search	Included in paper
Springer	"toxic comment classification with machine learning"	Between 2018 until 2022, English, Conference Paper,	102	[15]
	"toxic comment classification"		143	[3]
	"detecting cyberbullying using deep learning"		151	[15]
IJTSRD	"toxic comment classification"	-	-	[16]
IARJSET	"toxic comment classification"	-	-	[17]
MDPI	"toxic comment classification"	Between 2018 until 2022	4	[5]
AIP Conference Proceedings	"toxic comment classification"	Between 2018 until 2022	8	[10]
ACM	"deep learning toxic comment classification"	Between 2018 until 2022	92383	[23], [2], [26], [20], [7], [6], [8], [4],
Semantic Scholar	"toxic comment classification"	Between 2018 until 2022, Type Journal Article	2810	[21] (published in KDIR Conference)
SMU data science review	"toxic comment classification"	Between 2018 until 2022	4	[24]

III. RESULT & DISCUSSION

A. RQ1: Which data sets are utilized to detect toxic comments?

In this section, we are aiming to learn more about the kind of data sets that are utilized in the research paper. To obtain a better understanding, we have created a list of tables (Table III) that is organized by the year of release. We divided the table into different columns, such as, the name of the datasets, total samples, dataset sources and the output distribution.

Table III. *Details of the datasets and its output distribution*

Reference	Datasets	Samples	Source	Output Distribution
2018 [12], [23], [2], [10], [17]	Wikipedia's talk page edits.	159571	Kaggle	6 classifications (toxic, severe toxic, obscene, threat, insult, and identity hate).
2018 [19]	Collaborative Knowledge Repository	100000	Wikipedia	3 classifications (personal attack, racism, and sexism).
	Microblogging	16000	Twitter	
	Q&A forum	12000	Formspring	
2019 [11], [16], [15], [21]	Wikipedia's talk page edits.	159571	Kaggle	6 classifications (toxic, severe toxic, obscene, threat, insult, and identity hate).
2019 [8]	WikiDetox	160000	Wikipedia	2 classifications (toxic, non-toxic)
2019 [4]	Quora insincere Question Classification Challenge	1306122	Kaggle	2 classifications (sincere, insincere)
2020 [9]	Twitter Comments	56745	Github	2 classifications (toxic, non-toxic)
2020 [18]	Online Toxic Comments	95981	Kaggle	6 classifications (threat, insult, toxic, severe toxic, obscene, or identity hate).
2020 [14]	Jingdong Platform Octopus Collector	14831	Jingdong	3 classifications (positive, negative, medium)
2020 [7],[6]	Reddit Comments on r/AskReddit Figure eight crowdsourcing platform	10000	Reddit	2 classifications (toxic, non-toxic)
2020 [24]	Wikipedia talk pages	159571	Kaggle	2 classifications (toxic, non-toxic)

(Continued) Table III. *Details of the datasets and its output distribution*

Reference	Datasets	Samples	Source	Output Distribution
2021 [1]	IMDB Review	50000	IMDB	2 classifications (positive, negative)
	COVID19-Fake	10700	COVID-19 Website	2 classifications (positive, negative)
2021 [13]	Wikipedia	3000	Wikipedia	1 classification (attack)
	Twitter	3000	Twitter	2 classifications (racism, sexism)
	Formspring	3000	Formspring	1 classification (bully)
2021 [26]	Wikipedia talk pages	159571	Kaggle	6 classifications (toxic, severe toxic, obscene, threat, insult, identity hate)
	Twitter Stream API (Founta)	50425	Twitter	4 classifications (abusive, hateful, normal, spam)
	Twitter and Corpus (Waseem)	18625	Twitter, Corpus	4 classifications (racism, sexism, both, neither)
	Facebook Comments (Kumar)	14998	Waseem	3 classifications (non-aggressive, overtly aggressive, covertly aggressive)
2021 [20]	Wikipedia	143000	Wikipedia	6 classifications (safe comments, toxic, obscene, threat, insult, identity hate)
	Youtube Comments	3700000	Youtube	6 classifications (safe comments, toxic, obscene, threat, insult, identity hate)
2021 [5]	Wikipedia talk pages	159,571	Kaggle	6 classifications (toxic, severe toxic, obscene, threat, insult, identity hate)
2021 [3]	Jigsaw Toxic Comment Classification Challenge / Wikipedia talk pages	159,571	Kaggle	6 classifications (toxic, severe toxic, obscene, threat, insult, identity hate
	Hate Speech Dataset	N/A	Github	2 classifications (toxic, non-toxic)
	Sexual Abusive YouTube Comments	1000	Zenodo	2 classifications (toxic, non-toxic)
2022 [22]	The Unhealthy Comments Corpus (UCC)	44355	Github	8 classifications (antagonize, condescending, dismissive, generalization, generalization unfair, healthy, hostile, and sarcastic).

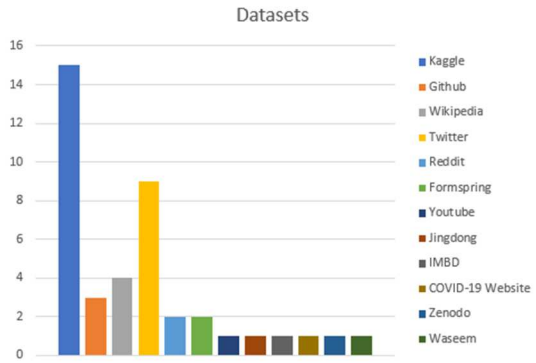


Fig 2. *Datasets used*

As shown in Fig 2, datasets that are being used from our references are mostly Kaggle, Twitter and Wikipedia. In a five years period, Wikipedia's talk page edits in Kaggle were the most popular datasets from five different researches. These top three sources are often used because of the large selection of datasets that can be used to help analyze the effectiveness of their algorithms in order to classify toxic comments.

B. RQ2: What are the most common deep learning models used to detect toxic comments?

In this section, we want to analyze more about the deep learning algorithm used by each research paper. To do so, we have created a table. Table IV provides the method used, the preprocessing and performance of the algorithms.

Table IV. *Machine Learning and Deep Learning Algorithm*

Method	Reference	Preprocessing	Performance (%)
CNN	[11]	GloVe	Accuracy: 97.27%
	[12]	FastText	F1-Score: 0.8675%
	[19]	SSWE	Precision: 93% Recall: 94% F1-Score: 93%
	[23]	Word2Vec	Accuracy: 91.2% Specificity: 91.7% False disc rate: 8.3% Recall: > 90% F1-Score: > 90% Precision: >90%
	[10]	N/A	Accuracy: 98.93%
	[17]	GloVe, Word2Vec	7 Epoch Accuracy: 98.04%
	[3]	Glove Embeddings (100d)	Accuracy: 95% F1 Score: 87%
LSTM	[9]	LabelEncoder, SpaCy	Accuracy: 94.94% Precision: 94.49% Recall: 92.79%
	[11]	GloVe	Accuracy: 97.01%
	[13]	Gradient Descent	Accuracy: 79.1%
	[19]	SSWE	Precision: 91% Recall: 93% F1-Score: 88%
	[7]	GloVe, Topic, Sent	ROC_AUC: 91% Accuracy: 82.5% Macro F1: 83%
	[6]	N/A	F1-Score: 81%
	[17]	Word2Vec	7 Epoch Accuracy: 98.77%
	[24]	N/A	Total Positive Rate: 67% F1: 73% Precision: 81% Recall: 66%
Bi-LSTM	[12]	FastText	F1-Score: 85.98%
	[19]	SSWE	Precision: 92% Recall: 95% F1-Score: 93%
	[4]	Topic, GloVe	F1-Score: 66.7%
GRU	[3]	Glove Embeddings (100d)	Accuracy: 96% F1 Score: 90%
		Fasttext Embeddings (300d)	Accuracy: 96% F1 Score: 90%
Bi-GRU	[12]	FastText	F1-Score: 86.43%
NN	[17]	TF-IDF	7 Epoch Accuracy: 98.91%
DNN	[8]	AutoML, LEAF	AUROC: 84.3%
LSTM-CNN	[11]	GloVe	Accuracy: 96.95%
	[22]	GloVe, TensorFlow	Average Micro F1: 88.76% Average Macro F1: 67.98% ROC_AUC: 71%
	[10]	CountVectorizer	Accuracy: 98.39%
Logistic Regression	[18]	Multi-Label Classification	Hamming-Loss: 2.43% Accuracy: 89.46% Log-loss: 2.14%
	[25]	Regex Tokenizer, PorterStemmer, NLTK, Bag of Words	Accuracy: 92.1% Precision: 95.9% Recall: 92.0% F1-Score: 93.9%
	[16]	OneVsRest classifier	Accuracy: 97.1%
	[10]	N/A	Accuracy: 91%
	[21]	FastText300+Udemy reviews word embeddings	Accuracy: 96%
	[21]	GloVe300+Udemy reviews word embeddings	Accuracy: 96%
Naive Bayes	[24]	N/A	Total Positive Rate: 48% F1: 64% Precision: 94% Recall: 48%
Multinomial Naive Bayes Classifier	[3]	TF-IDF	Accuracy: 94% F1 Score: 82%
SVM	[14]	Text deduplication	Accuracy: 87.24%
Multi-Task Multi-Embedding Deep Learning Architecture	[2]	TF-IDF, FastText, Glove, Our	F1-Score (negative labels clean comment): 99% F1-Score (positive labels illegal comment): 59%
Ensemble Model	[1]	Meta-classifier	Accuracy: 93.2%
	[20]	Glove, Word2Vec	F1-Score: 86.6%

(Continued) Table IV. *Machine Learning and Deep Learning Algorithm*

Method	Reference	Preprocessing	Performance (%)
BERT	[26]	N/A	Micro F1: 78.16% Macro F1: 63.72%
	[6]	N/A	F1-Score: 81%
	[5]	N/A	Accuracy: 98.6%
RoBERTa	[26]	N/A	Micro F1: 78.8822 Macro F1: 65.10%
XLM	[26]	N/A	Micro F1: 75.94% Macro F1: 51.22%
Six-headed Machine Learning Model	[15]	TF-IDF	Accuracy: 98.08%

In [19], it was shown that deep neural networks are much more effective in detecting cyberbullying among three datasets. Other research also stated that deep learning gave better results compared to machine learning models [13]. Therefore, we are focusing more on deep learning algorithms in the context of text processing. In deep learning, there are three main algorithms that are largely used, such as CNN, RNN, LSTM, GRU which will be explained in detail below.

Usually, a machine learning algorithm learns from a dataset it is given to do a specific task, it recognizes patterns from the data given and makes predictions for new data. Deep learning algorithms are a subset of machine learning algorithms that are more complex, these algorithms consist of layers of algorithm to make a logic structure to draw a conclusion of the data given. The amount of data needed is more but there is minimal human intervention.

An example of a machine learning algorithm is logistic regression. Logistic regression is a machine learning algorithm that is used to predict whether something is true or false, after the model is fed with the training data to form an S-shaped curved line called a logistic function, this line can then be used to predict new data samples.

Convolutional Neural Network (CNN) is a neural network that is usually used to detect and identify an image and has a unidirectional model structure. During the training process, the neural network is fed with training data and processed with random weights inside the hidden layers, after the process is done, we compare the prediction result with the actual result and calibrate the weights inside the hidden layers. What makes CNN different from other neural networks are the Convolutional Layers and Fully Connected Layer. Convolutional and pooling Layers are used to extract the features from the dataset, while the Fully-Connected layers are used to shape the data we received from the feature extraction into a vector.

Although CNN is mostly used in Computer Vision, recent studies have also used CNN in natural language processing which has unprecedented results [11], CNN creates a row of fixed-dimension vectors for each word, where it can be used to make n-grams that seem like a one-way node of various sizes being passed over the words.

In terms of accuracy, CNN itself can be categorized as an algorithm that is able to compete with other algorithms because of its high accuracy value where it can be seen in Table 4 that CNN itself can gain more than 85 percent. Furthermore, it can be improved by using word embeddings like GloVe and Word2Vec to around 90% of accuracy.

Long Short-Term Memory (LSTM) is a special kind of recurrent neural network that has internal memory capacity to store and remember extended sequences, which is beneficial for text classification because it can recall long-term memory compared to RNN itself [19]. LSTM creates each hidden node into a memory cell where it can store additional data. Another advantage is LSTMs have input and forget gates, which allow for greater control of the gradient flow and improved maintenance of long-range dependencies [24]. These gates are how LSTM solves the RNN's problems, where they suffer in vanishing gradients [11].

Nowadays, LSTM is the most frequently used algorithm in deep learning. As a result of their high performance and accuracy value. In [11], LSTM is able to gain approximately 97% of accuracy which beats the accuracy score of other machine algorithms alone. That's what makes LSTM very widely used, especially in the field of Natural Language Processing (NLP).

Gated Recurrent Unit (GRU) is a special kind of recurrent neural network that has a gating mechanism and internal memory capacity, the mechanism that decides what information is passed to the output and removes or forgets irrelevant information. The GRU is similar to LSTM because it also solves RNN's vanishing gradient problem.

Models that include GRU with word embedding are one of the most effective models and are able to compete with other models in accuracy, it is proven that GRU can get a 96% accuracy score using GloVe or FastTex .[3] that is the reason why GRU is a popular choice in the field of Natural Language Processing.

Hybrid Model is combining two or more separate methodologies in order to improve accuracy. When compared to machine learning approaches, CNN+LSTM has a higher F1-Score [22]. However, when compared to single models based on datasets, the accuracy may be lower. [11] shows that CNN+LSTM+GloVe is less accurate than CNN+Glove or LSTM+GloVe alone.

It indicates that, aside from the model itself, not every hybrid model can attain improved accuracy. Some studies necessitate a more thorough examination in order to choose the optimal classification approach.

IV. Conclusion

In conclusion, LSTM or Long Short-Term Memory is the most used and consistently yields better accuracy in classifying toxic comments compared to other models with the highest at 98.77% accuracy depending on the pre-processing. GRU or Gate Recurrent Unit is also commonly used and yields good accuracy. While for Logistic Regression, the highest accuracy is only at 96%. However, we found that using Neural Networks with TF-IDF pre-processing methods yields the best accuracy for this task. Many researchers also try to propose new models by

combining two or three different methods to get a more accurate classification. However, hybrid models sometimes have lesser accuracy than the original model.

For datasets, we found that the most frequently used datasets are from Kaggle. Specifically, Wikipedia's talk page edits datasets that have over 150000 samples with a total of 13 out of 26 research papers using this particular datasets. We can deduce that researchers are recommended to use Wikipedia's talk page edits dataset from Kaggle in order to be able to accurately deduce the difference in evaluation scores between each algorithm. Furthermore, future studies might include more hybrid models with different models which have not been covered in this paper and researchers could use LIME model interpretability techniques to further help in finding the most reliable models rather than only depending on the evaluation metrics score for choosing the best model.

REFERENCES

[1]. A. Mohammed and R. Kora, "An effective ensemble deep learning framework for text classification," *Journal of King Saud University - Computer and Information Sciences*, 2021.

[2]. A. Elnaggar, B. Waltl, I. Glaser, J. Landthaler, E. Scepankova, and F. Matthes. "Stop Illegal Comments: A Multi-Task Deep Learning Approach." *In Proceedings of the 2018 Artificial Intelligence and Cloud Computing Conference (AICCC '18). Association for Computing Machinery, New York, NY, USA, 41–47, 2018*

[3]. A.Mahajan, D. Shah, and G. Jafar, "Explainable AI approach towards toxic comment classification." In Emerging Technologies in Data Mining and Information Security (pp. 849-858), 2021.

[4]. D. Y. Kim, X. Li, S. Wang, Y. Zhuo, and R. K. Lee. "Topic enhanced word embedding for toxic content detection in Q&A sites." *In Proceedings of the 2019 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM '19). Association for Computing Machinery, New York, NY, USA, 1064–1071, 2019*

[5]. Fan, W. Du, A. Dahou, A. A. Ewees, D. Yousri, M. A. Elaziz, A. H. Elsheikh, L. Abualigah, and M. A. A. Al-qaness, "Social media toxicity classification using Deep Learning: Real-World Application UK Brexit,"(MDPI), 2021.

[6]. H. Almerekhi, H. Kwak, J. Salminen, and B. J. Jansen. "Are These Comments Triggering? Predicting Triggers of Toxicity in Online Discussions." *Proceedings of The Web Conference 2020. Association for Computing Machinery, New York, NY, USA, 3033–3040, 2020*

[7]. H. Almerekhi, Supervised by B. J. Jansen, and co-supervised by H. Kwak. "Investigating Toxicity Across Multiple Reddit Communities, Users, and Moderators." *Companion Proceedings of the Web Conference 2020. Association for Computing Machinery, New York, NY, USA, 294–298, 2020*

[8]. J. Liang, E. Meyerson, B. Hodjat, D. Fink, K. Mutch, and R. Miikkulainen. "Evolutionary neural AutoML for deep learning." *In Proceedings of the Genetic and Evolutionary Computation Conference (GECCO '19). Association for Computing Machinery, New York, NY, USA, 401–409, 2019*

[9]. K. Dubey, R. Nair, M. U. Khan, and P. S. Shaikh, "Toxic comment detection using LSTM," *2020 Third International Conference on Advances in Electronics, Computers and Communications (ICAECC)*, 2020.

[10]. M. A. Saif, A. N. Medvedev, M. A. Medvedev, and T. Atanasova, "Classification of online toxic comments using the logistic regression and neural networks models," ,2018.

[11]. M. Anand and R. Eswari, "Classification of abusive comments in social media using Deep Learning," *2019 3rd International Conference on Computing Methodologies and Communication (ICCMC)*, 2019

[12]. M. Ibrahim, M. Torki, and N. El-Makky, "Imbalanced toxic comments classification using data augmentation and Deep Learning," *2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA)*, 2018.

[13]. M. Mahat, "Detecting cyberbullying across multiple social media platforms using Deep Learning," *2021 International Conference on Advance Computing and Innovative Technologies in Engineering (ICACITE)*, 2021.

[14]. M. Zhang, "E-Commerce Comment Sentiment Classification Based on Deep Learning." *IEEE 5th International Conference on Cloud Computing and Big Data Analytics (ICCCBDA), 2020, pp. 184-187, 2020*

[15]. N. Chakrabarty1, "A Machine Learning Approach to Comment Toxicity Classification." , 2019

[16]. P. Ravi, H. N. Batta, S. Greeshma, S.Yaseen, "Toxic Comment Classification" , 2019

[17]. R.Sharma , & M.Patel, "Toxic Comment Classification Using Neural Networks and Machine Learning" - IARJSET. Academia, 2018

[18]. Rahul, H. Kajla, J. Hooda, and G. Saini, "Classification of online toxic comments using machine learning algorithms," *2020 4th International Conference on Intelligent Computing and Control Systems (ICICCS)*, 2020.

[19]. S. Agrawal and A. Awekar, "Deep learning for detecting cyberbullying across multiple social media platforms," *Lecture Notes in Computer Science*, pp. 141–153, 2018.

[20]. S. Alshamrani, A. Abusnaina, M. Abuhamad, D. Nyang, and D. Mohaisen. "Hate, Obscenity, and Insults: Measuring the Exposure of Children to Inappropriate Comments in YouTube." *In Companion Proceedings of the Web Conference 2021 (WWW '21). Association for Computing Machinery, New York, NY, USA, 508–515, 2021*

[21]. S. Carta, A. Corriga, R. Mulas, D. Recupero, and R. Saia, "A Supervised Multi-class Multi-label Word Embeddings Approach for Toxic Comment Classification," *Proceedings of the 11th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management*, 2019

[22]. S. Gilda, L. Giovanini, M. Silva, and D. Oliveira, "Predicting different types of subtle toxicity in unhealthy online conversations," *Procedia Computer Science*, vol. 198, pp. 360–366, 2022.

[23]. S. V. Georgakopoulos, S. K. Tasoulis, A. G. Vrahatis, and V. P. Plagianakos. "Convolutional Neural Networks for Toxic Comment Classification." *In Proceedings of the 10th Hellenic Conference on Artificial Intelligence. Association for Computing Machinery, New York, NY, USA, Article 35, 1–6, 2018*

[24]. S. Zaheri, J. Leath, and D. Stroud, "Toxic Comment Classification" , SMU Data Science Review: Vol. 3: No. 1, Article 13, 2020

[25]. V. Jain, V. Kumar, V. Pal, and D. K. Vishwakarma, "Detection of cyberbullying on social media using machine learning," *2021 5th International Conference on Computing Methodologies and Communication (ICCMC)*, 2021.

[26]. Z. Zhao, Z. Zhang, and F. Hopfgartner. "A Comparative Study of Using Pre-trained Language Models for Toxic Comment Classification." *In Companion Proceedings of the Web Conference 2021 (WWW '21). Association for Computing Machinery, New York, NY, USA, 500–507, 2021*

Toxic Speech Detection

Animesh Koratana & Kevin Hu
Stanford University
Department of Computer Science
{koratana,huke}@stanford.edu

Abstract

Due largely to the ubiquitous rise of social media over the last decade, the mode of communication that people subscribe to has significantly changed. While it's allowed for a more connected, and informed world, it has also made room for a new phenomenon: toxic speech. An open platform for the world to produce, comment, and share content has opened has allowed rather opportunistic users to participate in hate speech: peddling sexist, racist, xenophobic, and all around negative comments. Our project is motivated by this trend. We aim to develop high accuracy classifiers on a hate speech datasets using modern deep learning techniques to primarily identify the existence of hate speech in comments and texts in an efficient manner. We explore the use of various approaches to this classification problem, including using logistic regression, CNN, and RNN based models. We also evaluate their inference efficiency and proceed to propose and test a cascade model that achieves high throughput in the average case while maintaining a high accuracy by combining multiple approaches.

1 Introduction

With the rapid growth of online platforms and forums, people have become increasingly aware of the problem of abusive language and hate speech. When an individual decides to engage in an online discussion on social media, blogs, or comment sections, they are exposing themselves to the risk of being harassed by trolls or rude commenters. Instances of offensive comments are quite common and have negatively impacts the dynamics of the online community as well as the user experiences of the targeted individuals. This phenomenon, in addition to the supposed structured misinformation bias 2016 election, has led to new vigor in the field of toxic comment classification. As such, most companies that run internet platforms filters their sites by deploying a combination of human moderators and automated abusive language detection algorithms. Our project is motivated by this trend. Through our project, we intend to build and evaluate a toxic speech classifier using a few of the state-of-the-art, deep learning approaches. To name specific leading text classification architectures, we intend to implement and evaluate the ability of a Convolutional Bidirectional GRU with Attention layers and Very Deep Convolutional Neural Networks (VDCNN) (Conneau et al. [2016]) to classify and detect hate speech efficiently.

Many of the simple abusive language detection systems use regular expressions and a blacklist (which is a pre-compiled list of offensive words and phrases) to identify comment that should be removed. However, the problem with these models – and ultimately the difficulty of this task – is that hate speech is more than keyword recognition and pattern identification in the grammar. For instance, one can get around the system by removing spaces between words, use alternative spellings, or create homonyms that will still make sense to humans but cannot be detected by machines. The task, then, as suggested by the title of the paper, is to build an accurate hate speech detection using a deep neural network. We define hate speech from the definition established by Davidson et al. [2017] as *any*

32nd Conference on Neural Information Processing Systems (NIPS 2018), Montréal, Canada.

communication that disparages a person or a group on the basis of some characteristic such as race, color, ethnicity, gender, sexual orientation, nationality, religion, or other characteristic.

Recent trends in machine learning and deep learning in the context of natural language processing have yielded significant improvements to the ability of machines to detect subtleties in natural language, including hate speech. However, these speedups often come at the cost of speed and scalibility. Modern deep learning architectures have millions of parameters with modern architectures such as VDCNNs approaching close to 8M parameters (Conneau et al. [2016]). Using hate speech detectors in practice often bears a significant challenge of implementation. As modern deep learning architectures follow the general trend of increasing representational power with depth, companies are forced to be more cautious about their deployed architectures.

In this project we intend to explore both these arenas together. We will begin by exploring a robust baseline approach to classification using logistic regression. We then will proceed to make our novel contribution by evaluating an attention-based RNN, and CNN based classifier on the same dataset. We finally will evaluate standard classification approaches on the task of toxic comment classification, and explore their computational efficiencies.

2 Related Work

Existing works in this domain have tackled the problem of toxic speech detection as a classification task, which is the logical approach the goal is to either performing binary classification to identify whether a comment is toxic, or performing multi-class classification to identify the type of toxic comment. Among the variants of hate speech detection models, they can be separated into what Zhang et al. called the "classical methods" and deep learning methods. Specifically, the classical methods are ones that require manual feature engineering, and include Logistic Regression, Bayesian models, SVM, and random forest. On the other hand, deep learning methods use different neural structures to learn abstract, high-level features from the training data.

The classical methods have proven to be very successful in various circumstances, but also have its shortcomings. In particular, the feature engineering process takes enormous effort and is always inexhaustive, as people can easily warp their messages and find ways around being detected by the system. Some of the most common features include word vectors, character n-grams, lexical features (based on the assumption that hateful speeches contain negative or offensive words), linguistic features (such as POS tags), and knowledge-based features (like the relative appearance of "kill" and "Jews") which cannot be easily constructed and is always evolving. We think that while classical approaches have their merits, there is great potential for deep learning models in this field because its feature extraction is automated and can perhaps capture patterns and trend that humans cannot think of. As a result, logistic regression is a fitting choice of baseline. Not only is it simple to implement, it gives us a general idea of the performance of these classical methods which we are trying to outperform.

The two models that we implemented and tested in this project are inspired by existing text classification models but are slightly different from the ones that are tailored to this specific test. The idea of a VDCNN initially came to us from the success of CNNs in image processing. The particular model that we implemented is based on Conneau's VDCNN for text classification. A problem with the method is that it is character-based – as we increase the depth of the VDCNN, the training time increases exponentially, which we cannot fully test due to limitations in computational power. As such, we modified Conneau's model into a word-based model and used pre-trained word embedding from FastText. This drastically improves the training time at the sacrifice of the number of features (words instead of chars), and makes the project more manageable. Additional deep learning approaches include Zhang et al.'s use of a combination of CNN and GRU and Founta et al.'s use of a combination of text data (using RNN) and the online user's metadata to predict toxic comments. Given the success of RNN variants in "interpretting" meanings in text and its use in the various related works, we decided to implement a bi-directional GRU with attention to tackle the problem of hate speech detection.

3 Approach

For this project, we take two deep learning approaches to solve the toxic speech classification problem. The first model that we are using is a GRU RNN with attention, which is based off of the state-of-

the-art LSTM and RNN paradigms, but uses specific modifications. The second approach is using a VDCNN (Very Deep Convolutional Neural Network), which is motivated by the success of very deep networks in computer vision, and based on the theory that deeper networks can encapsulate more information and achieve higher test accuracy. We have chosen two commonly used baseline models for NLP text classification: logistic regression with word and char n-gram features.

3.1 Baselines

Most of the recent papers in text classification and hate speech detection uses efficient, linear models as baselines. Badjatiya et al. uses logistic regression as a strong baseline to evaluate for text classification tasks.

The logistic regression baseline uses our familiar gradient descent update rule:

$$\theta_j := \theta_j + \frac{\alpha}{m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)}) x_y^{(i)}$$

What's more interesting is the process of feature extraction. The features that we use include both the words, as well as character n-grams (where $2 \leq n \leq 6$) present in the training data. Moreover, we weigh the features by applying a TF-IDF (term frequency – inverse document frequency) transform, which is a measure of the importance of each of the words and n-grams in the corpus. The TF-IDF score of a feature is calculated by multiplying its TF (term frequency) and IDF (inverse document frequency), both of which can have various interpretations. Our particular baseline uses the feature vectors extracted by TfidfVectorizer from the sklearn package. The baseline was implemented and tested by us.

3.2 GRU & LSTM with Attention

One type of model that we will be testing is an rnn based model. Specifically we will be testing variants of GRU and LSTM based models. Variants of the RNN have proven to be successful in many NLP tasks and we think that it will perform well with classification tasks. The current model that we implemented is showing promising results and will be further tuned for the final report.

Each comment is represented as a sequence of words, which is padded to the maximum length of the sentences in the training data. To reduce the time of training word embeddings, we start by looking up 300-dimensional vector representations using the FastText library. Then we pass through the word embeddings through a bi-directional GRU or LSTM to obtain the sentence embedding.

The embedding is then fed into a scaled dot product attention layer. This is slightly different from the dot product attention that we learned in class by incorporating a scale factor into the calculation of the attention scores. Specifically, we take the current decoder state to be the query q and take each of the encoder hidden states to be the keys. For each of the keys k_i, we calculate the attention score with the following formula:

$$\text{Attention Score} = \frac{q^T k_i}{\sqrt{d_k}}$$

where d_k is the dimension of the key vectors. The motivation of using scaled dot product attention is that the value of the dot product becomes very large with higher dimensions, which leads the softmax function to go into regions with exceptionally small gradients. Scaling the attention score by a factor of $\frac{1}{\sqrt{d_k}}$ will allow us to avoid or at least alleviate the issue (Vaswani et al.).

Its result of the model is mapped into a n-dimensional output space (n = 7 is the number of classes, including "clean", "toxic", "severe toxic", "obscene", "threat", "insult", and "identity hate") through a fully connected linear decoder and converted into probabilities using the softmax function. The model is trained using CE loss and SGD optimizer. We implemented this model by hand and from scratch with the exception of the accuracy computing function, which was adapted from PyTorch's default ResNet training code [1].

[1] https://github.com/pytorch/vision/tree/master/torchvision/datasets

3.3 VDCNN

Our VDCNN references the architecture of Conneau et al. which has proven to be successful in the field of text processing. In the future, we will be looking to test different configurations and adjustments to the model in order to improve performance; but for now, we are starting by writing our own implementation of VDCNN-9 as noted in the paper. The architecture is as followed:

Each text comment is represented as a sequences of characters, which is either truncated or padded to a fixed length of 1024. For each char, we look up an embedding of size 16. The embeddings go through a 1D convolutional layer with window size 3 and 64 output channels. Now the data matrix has size 64×1024, and we will pass it through 4 covoluntional blocks, each containing two 1D convolutional layers. The idea of each convolutional block is to double the output channel after which a pooling layer will reducing the number of embeddings by half through downsampling. This prevents the exponential growth of the number of parameters and keeps the memory usage consistent. The internal structure of a convolutional block with input dimension D is: Conv1d(D, 2D, 3) –> Batch Norm –> ReLU –> Conv1d(2D, 2D, 3) –> Batch Norm –> ReLU (the third parameter of Conv1d is window size). Note that the four convoluntional blocks contains 8 convolutional layers; adding the initial convolutional layer, we reach a total depth of 9. After the convolutional blocks, the output is of size 512×128. We now apply a k-max pooling layer with $k = 8$ to reduce the dimension to 512×8. This is flattened into a vector with size 4096 and passed through three linear layers in this order: Linear(4096, 2048) –> ReLU –> Linear(2048, 2048) –> ReLU –> Linear(2048, n), where n is the number of classes. Currently, we have set $n = 2$, meaning that either a comment is toxic or clean; but we can create more categories for the text labels such as "insult", "threat", and "identity hate" comments. Finally, we put the output through the softmax function to predict the probability that a comment belongs to each class:

$$P(\text{Class} = x_j) = \text{Softmax}(x_j) = \frac{\exp(x_j)}{\sum_{i=1}^{n} \exp(x_i)}$$

The model is trained using CE loss and Adam optimizer. We implemented this model by hand so that we could adapt it to our use cases.

4 Experiments

4.1 Data

In 2017 Google Jigsaw published a dataset on Kaggle labeled "Toxic Comment Classification Challenge". The dataset includes 223,549 user comments, annotated with labels that fall into one or more of the following categories: clean, toxic, obscene, insult, identity hate, severe toxic, and threat [2]. The dataset is from a real world example, and a majority (about 200,000) of the samples in the dataset fall under the "clean" label. Our classifier's intent is to be able to distinguish between these different categories at two levels of granularity.

The task we are going to attempt is to construct a binary classifier to distinguish between toxic speech and clean speech. For this we will consider all elements that fall under any one of the categories of toxic, obscene, insult, identity hate, severe toxic, and threat as toxic speech and clean as the other label.

The dataset itself provides a standardized test set to compute accuracies on. However to account for overfitting, we wil split our training set randomly into an 80/20, test/dev split. The evaluation of the model on the dev split at each epoch will be used to choose the final model that will then be tested on the test set.

4.2 Evaluation Method

We evaluate our models and approaches on two key criteria: the F1 and test accuracy. The F1 score is a measure of the models accuracy as it is a harmonic average of precision and recall. We compute the F1 score for each of the outputted categories and average the F1 scores for each category to compute an composite F1 score to represent the overall precision and recall of the system. The F1 score is a

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

standard measurement used for NLP classification tasks, which we can use to compare with existing models. The second evaluation method we use is the accuracy of the model's predictions on the test set. This is our stronger metric to test as it gives us a simple and direct measure of the ability of our models to detect toxic comments. Together the F1 score and test accuracy paint a good picture about the performance of the models.

4.3 Experimental Details

4.3.1 Architectures

Our baseline model consists of a logistic regressor trained on word embeddings. This is a simple method that takes as input the average of the word embeddings and outputs a probability of a specific class membership.

We consider two deep learning/ neural network based approaches to solving our task: VDCNN (Conneau et al. [2016]) and GRU/LSTM based method (Chung et al. [2014]).

We initialize a VDCNN-9 as constructed in the paper with maxpooling blocks, and no residual connections. VDCNN has an initial convolutional layer and four blocks of convolutional layers (each block has the same number, but vary between types of VDCNNs, similar to ResNets). Thus, a VDCNN-9 has an initial convolutional layer and two convolutional layers in each subsequent block. We use the standard VDCNN (Conneau et al. [2016]) on the Jigsaw dataset.

We also initialize a bidirectional GRU and LSTM with attention as an alternate classifier for our dataset (Chung et al. [2014], Lin et al. [2017]). The rnn-based model has a hidden size of 500, and two layers. The loss function and gradient steps are evaluated as detailed above. The bidirectional GRU/LSTM takes as input the word embeddings, and its output as a "sentence embedding". The sentence embedding is then fed through the attention layer as detailed by Lin et al. [2017] and then fed into a fully connected classifier.

We modify both networks to have the option to classify on word embeddings instead of character embeddings and thus give ourselves the option to train faster and also use a consistent source of embeddings across trials and architectures. The pretrained word embeddings we use are 300 dimensional word vectors trained on Wikipedia 2017, UMBC webbase corpus and statmt.org news dataset (16B tokens, 1M word vectors) (Mikolov et al. [2018]). These embeddings also allow us to use sub-word embeddings to generate embeddings from words that would otherwise be classified as unk in the dataset.

Each of these networks and approaches are implemented and executed using Pytorch 1.0.1.

4.3.2 Hyperparameters

For all experiments we are running with VDCNN, we use a batch size of 128, SGD with a momentum of 0.9 and weight decay of 1e-4. Similar to the paper we start with a learning rate of 0.01 for 15 epochs with milestones for learning rate decay of a factor of 10 at epochs 3,6,9,12, and 15.

For the GRU and LSTM based models, we use a batch size of 64, SGD with a momentum of 0.9 and weight decay of 1e-4 with gradient clipping. We start with a learning rate of 0.001 with a decay by factor of 10 on plateaus.

5 Results

Our experiments evaluated the performance of the described models on our task with many variants. As mentioned for each model, we computed the F1 score and the accuracy of the model on the test set in the binary classification setting.

We note that when it comes to accuracy, the Bi-LSTM with attention and pretrained embeddings seems to be the top contender with a test accuracy score 0.989. The LSTM adn GRU with attention and FastText embeddings also seems to achieve the highest F1 score above the other three trials.

	F1 Score	Accuracy
Baseline: Logistic Regression	0.44	0.967
Bi-GRU	0.57	0.971
Bi-GRU (FastText Embeddings)	0.61	0.974
Bi-GRU (Attention + FastText Embeddings)	0.66	0.987
Bi-LSTM	0.60	0.975
Bi-LSTM (FastText Embeddings)	0.62	0.980
Bi-LSTM (Attention + FastText Embeddings)	**0.66**	**0.989**
VDCNN-9 (FastText Embeddings)	0.62	0.975
VDCNN-17 (FastText Embeddings)	0.62	0.978

5.1 Pretrained Embeddings

A key observation is the effect of the use of pretrained, FastText embeddings on the final computed scores. We find that using pretrained word embeddings actually boosts the accuracy of the trained models by a nontrivial amount. We can see this effect by contrasting the accuracies of the Bi-GRU/LSTM and the Bi-GRU/LSTM with pretrained embeddings (Figure 1). The addition of embeddings seems to give a substantial boost in accuracy to the underlying models.

We believe these pretrained embeddings offer a significant boost in accuracy for two central reasons:

1. The FastText pretrained embeddings are able to calculate subword embeddings. With a manual survey of our dataset we noticed that the frequency of tokens was relatively small, with about 38A further inquiry into this show that tokens like "sucklol" and "f**kings**t" (sampled from our dataset) are counted as discrete tokens. This is not a surprise due to our dataset being sourced from a number of social media forums. Our data preprocessing removes these tokens from the vocabulary.

   By adding the pretrained embeddings by FastText, we are able to compute embeddings for these types of tokens, encoding more of the meaning within the post.

2. The pretrained embeddings are also trained on an extremely large corpus. In fact the embeddings are trained on 16B tokens, 1M word vectors (Mikolov et al. [2018]), which is significantly larger than the training set we have available to us.

5.2 Attention

We notice that the addition of scaled dot-product, self-attention on the GRU or LSTM gives marginal increases in accuracy (Figure 1). For example the addition of attention boosts the test accuracy by about a 1.3We postulate that this boost is due largely to the ability of attention to prioritize specific parts of the sentence over other parts. This may be particularly helpful in this classification task because in identifying hate speech, only a few parts of the sentence may be necessary. For example in the sentence "go home you idiot", the part of the sentence that matters is the negative assertion ("idiot") and the part of the sentence that makes it pointed to a person or group ("you").

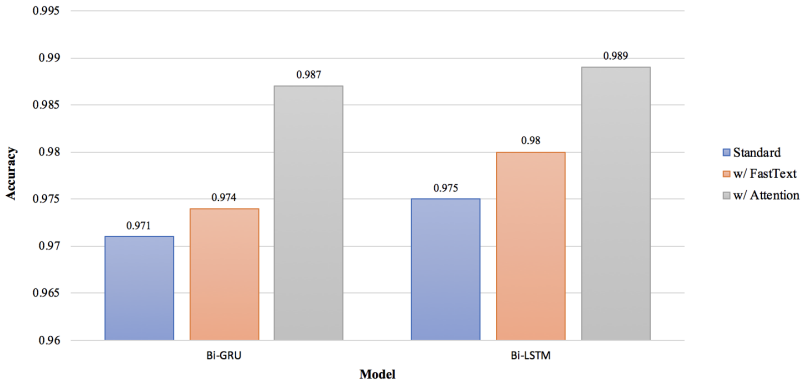


Figure 1: Effects of pre-trained embeddings and attention on test accuracy

5.3 Inference Speeds

One of the largest barriers to using deep NLP methods in industry is the computational cost associated with deep learning. To get a sense of the magnitude of this cost, we ran a few experiments to estimate the inference speed of each. We use the same preprocessing on each of these models (with FastText embeddings) and evaluate a forward pass with one document for 10 trials of 100,000 runs each. With this we calculate the average runtime and the standard deviation of that runtime.

	Inference Speed (ms)	**St. Dev.**
Baseline: Logistic Regression	2.03 ms	0.02 ms
Bi-GRU (Attention + FastText Embeddings)	22 ms	1.02 ms
Bi-LSTM (Attention + FastText Embeddings)	28 ms	1.19 ms
VDCNN-17 (FastText Embeddings)	26 ms	0.65 ms

Each trial was controlled on the same machine with no other processes substantial processes running. The tests were run on a 6 core i7 Intel CPU.

As shown in Figure 2, the logistic regression model actually has a latency of 2.03 ms which is about 11x faster than its RNN and CNN based counterparts. This does not come as a surprise at all due to the extremely efficient and small computation required to run the forward pass of a logistic regressor compared to a 17 layer CNN or a RNN.

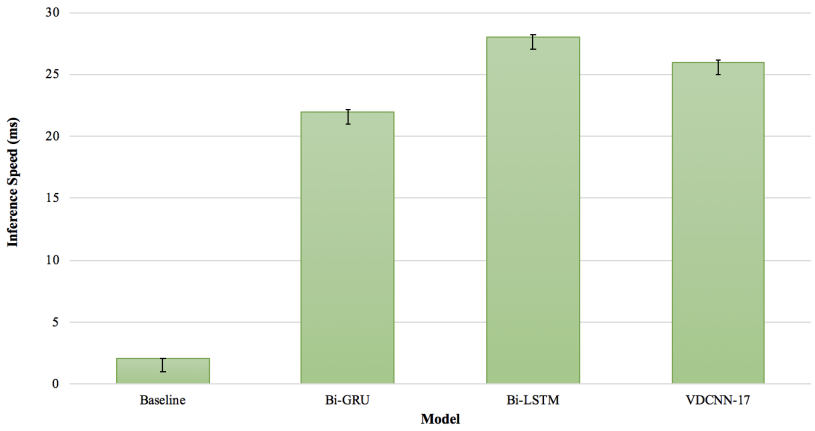


Figure 2: Inference speed of our deep learning models compared to baseline

5.3.1 Cascading Model

As one of our future directions, we propose a simple cascading model that combines multiple of our tested models to optimize for accuracy and speed in the average case. Cascading models use intermediate steps and confidence scores at each step. Each subsequent step has a more substantial computation cost. We test a simple cascading model composed of a logistic regression as the first step.

If the logistic regressor outputs a value between 0.3 and 0.7 (as a simple heuristic for the logistic regressor being unsure about the outputs), we feed the original input into the Bi-LSTM with attention and pretrained model (see Figure 3 below). On the test set, this cascading model gives us substantial speedups with an average latency of **5.18 ms**, still almost 6 times faster than using the Bi-LSTM w/ attention and embeddings alone. Only about 31% of the documents from the test set had a mediocre score on the logistic regressor and had to be pushed to the LSTM.

This also boosts the accuracy of the model to a higher accuracy than that of only a logistic regressor to **0.973**. This shows a promising result for the detection of hate speech in an efficient and scalable manner.

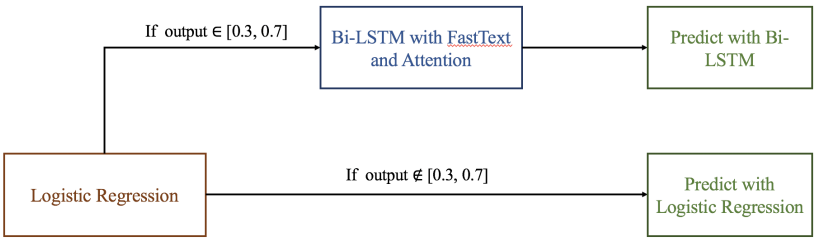


Figure 3: Diagram of a simple cascading model

6 Conclusion

From our experiments, we have shown that deep learning models can perform very well in the task of toxic comment detection. Both the Bi-GRU/LSTM and VDCNN models were able to produce higher F1 and accuracy compared to a fairly strong (albeit simple) logistic regression baseline. We have addressed the issue of computational time and cost for deep learning models by analyzing their inference speeds. Indeed, a huge disadvantage of CNNs and RNNs are there sluggish training and testing time compared to "classical methods." As such, we propose adopting a cascading model, which is made possible by the solid performance of time-efficient baseline models. The idea is that if a fast model like logistical regression or SVM is sufficiently confident in its prediction, then we use it; otherwise, we leave the decision to our deep learning model, which takes longer but is more accurate.

A limitation of our project is that our models are trained on Google Jigsaw's toxic comment dataset, which contains mostly long comments and seven types of toxic labels. As a result, our models are tailored to the dataset and can predictions for each of the seven categories. While we achieved high accuracy on our test set, we may not be able to achieve the same level of performance, if, say, the model is applied to tweets. In general, it is difficult to find sizable, cleanly labelled datasets for toxic comments, so we would have to compile and preprocess much more data if we were to build on this project. Moreover, we were not able to run a thorough analysis on the effect of depth in VDCNNs because the training time grows exponentially and we were only able to train up to a VDCNN-17. Ideally, we would like increase the depth up to 49 or higher to see if deep networks can further boost the accuracy.

For future work, there are many models and combination of deep learning paradigms that we would like to test and explore such as combining convolutional and recurrent neural networks, as well as state-of-the-art SVM models and feature extraction mechanisms used for toxic speech detection. We think that the cascading model shows promise. While we only implemented a simple version, it would be interesting to further optimize the criteria to determine the confidence of the baseline and to improve both the efficiency and accuracy of the model.

7 Code

This entire project's codebase can be found here: https://goo.gl/GgPnG1

References

Junyoung Chung, Çaglar Gülçehre, KyungHyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *CoRR*, abs/1412.3555, 2014. URL `http://arxiv.org/abs/1412.3555`.

Alexis Conneau, Holger Schwenk, Loïc Barrault, and Yann LeCun. Very deep convolutional networks for natural language processing. *CoRR*, abs/1606.01781, 2016. URL `http://arxiv.org/abs/1606.01781`.

Thomas Davidson, Dana Warmsley, Michael Macy, and Ingmar Weber. Automated hate speech detection and the problem of offensive language. In *Proceedings of the 11th International AAAI Conference on Web and Social Media*, ICWSM '17, pages 512–515, 2017.

Zhouhan Lin, Minwei Feng, Cícero Nogueira dos Santos, Mo Yu, Bing Xiang, Bowen Zhou, and Yoshua Bengio. A structured self-attentive sentence embedding. *CoRR*, abs/1703.03130, 2017. URL `http://arxiv.org/abs/1703.03130`.

Tomas Mikolov, Edouard Grave, Piotr Bojanowski, Christian Puhrsch, and Armand Joulin. Advances in pre-training distributed word representations. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC 2018)*, 2018.