Journal of Computational Social Science (2023) 6:1081–1101
https://doi.org/10.1007/s42001-023-00224-9

RESEARCH ARTICLE

Transfer learning for hate speech detection in social media

Lanqin Yuan[1] · Tianyu Wang[2] · Gabriela Ferraro[2] · Hanna Suominen[2,3] · Marian-Andrei Rizoiu[1,2]

Received: 27 September 2022 / Accepted: 1 September 2023 / Published online: 17 October 2023

Abstract

Today, the internet is an integral part of our daily lives, enabling people to be more connected than ever before. However, this greater connectivity and access to information increase exposure to harmful content, such as cyber-bullying and cyber-hatred. Models based on machine learning and natural language offer a way to make online platforms safer by identifying hate speech in web text autonomously. However, the main difficulty is annotating a sufficiently large number of examples to train these models. This paper uses a transfer learning technique to leverage two independent datasets jointly and builds a single representation of hate speech. We build an interpretable two-dimensional visualization tool of the constructed hate speech representation—dubbed the Map of Hate—in which multiple datasets can be projected and comparatively analyzed. The hateful content is annotated differently across the two datasets (racist and sexist in one dataset, hateful and offensive in another). However, the common representation successfully projects the harmless class of both datasets into the same space and can be used to uncover labeling errors (false positives). We also show that the joint representation boosts prediction performances when only a limited amount of supervision is available. These methods and insights hold the potential for safer social media and reduce the need to expose human moderators and annotators to distressing online messaging.

Keywords Hate speech · Transfer learning · Visualization · Twitter · Domain adaptation · Offensive speech

Introduction

Ubiquitous access to the internet brought with it profound change to our lifestyle: information and online social interactions are at our fingertips; however, it brings new challenges, such as the unprecedented liberalization of hate speech. Hate speech is a type of online abuse defined as "public speech that expresses hate or encourages violence toward a person or group based on race, religion, sex, or

Extended author information available on the last page of the article sexual orientation" [6]. Hate speech proliferation is suspected to be an important culprit in creating a state of political violence and in exacerbating ethnic violence, such as the Rohingya crises in Myanmar [40]. Hate speech has also been linked to its victims' health deterioration; several studies confirmed that racism and sexism are associated with poorer mental health, including depression, isolation, anxiety, and poor self-esteem [5, 32]. Considerable pressure is mounting on social media platforms to timely detect and eliminate hate speech, alongside cyber-bullying and offensive content [48].

This work addresses three open questions about detecting hateful content (i.e., hate speech, racist, offensive, or sexist content). The first question concerns constructing a more generic detection system for textual hate content. There is a considerable amount of work on detecting hate speech [8, 9, 11, 12, 44, 47]; however, many works rely on hand-crafted features, user information, or platform-specific meta-data, which limits its generalization to new data sets and data sources. The first question is can we design a *general-purpose* hate embedding and detection system which does not rely on expensive hand-crafted features and can adapt to a particular learning task? The second question relates to data availability. It is difficult to build large-scale datasets of online hate speech, as platforms usually report and remove such content. While this has the advantage of protecting the users, it also hampers researchers' efforts to build datasets. When such datasets are built, they are often of small scale and not representative of the entire spectrum of hateful content. The question is can we leverage multiple smaller, unrelated data sets to learn jointly and to transfer information between apparently unrelated learning tasks? The third question relates to the interpretation and analysis of hate speech by asking can we construct a tool for separating types of hate speech and characterizing what makes particular language hateful?

This paper addresses the above three open questions by jointly leveraging two unrelated hate speech data sets. We address the first two open questions by proposing t-HateNet, a novel neural network transfer learning pipeline. We train the system by predicting, for each dataset, whether a text is hateful or not. The system contains shared components that construct the textual embeddings and dataset-specific components that predict the final labels. We use pre-trained word embeddings, that we adapt to the current learning tasks using a *bidirectional Long Short-Term Memory* (bi-LSTM) [16] unit. This creates a single representation space capable of successfully embedding hateful content for multiple learning tasks. We show that the system can transfer knowledge from one task to another, thus boosting performance. The system operates solely on the analyzed text, making it platform and source independent. Given a new learning task, the hateful embeddings trained on other tasks can be directly applied; if labeled data is available for the new task, it can be used to contextualize the embeddings. We address the third open question by building the *Map of Hate*, a two-dimensional representation of the hateful embeddings described above. We use this embedding for several tasks. First, we visually show that hate classes are more separable using the joint embeddings constructed by t-HateNet than if trained individually. Second, the *harmless* classes across the two datasets—containing texts not labeled as hate speech—overlap in the joint representation. Finally, we use the Map of Hate to diagnose mislabeled examples and uncover systematic labeling issues.

The main contributions of this work are as follows: First, we assemble *HateNet*—a deep neural network architecture—capable of creating task-specific word and sentence embeddings without the need for expensive hand-crafted features. Second, we propose *t-HateNet*, which connects the HateNet architecture with *transfer learning* methods that allow leveraging several smaller, unrelated data sets to construct a general-purpose hate speech embedding. Third, we introduce the *Map of Hate*—an interpretable 2D visualization of hateful content, capable of separating different types of hateful content and explaining what makes text hateful.

Background and prerequisites

Despite its recency as a research field [13], online abuse has received considerable attention; its subproblems range from quantifying bias and stereotypes in detection models [3], to building dedicated web applications for actively reporting hate speech [23], to organizing shared tasks for aggression identification [22], to visualizing the geographical spread of hate speech [7], and building novel formulations of the hate speech type identification problem using fuzzy ensemble approaches [25]. In our presentation of related works, we concentrate on hate speech detection in the English language (”Hate speech detection”), hate speech mapping, visualization and interpretation (“Hate speech mapping, visualization and interpretation”), and domain adaptation (“Domain adaptation”).

Hate speech detection

Feature-based classifiers. Early approaches from 2015 to 2017 mainly used hand-crafted features together with simple off-the-shelf classifiers. For example, Waseem and Hovy [45] used a *Logistic Regression model* with *character-level features* to classify tweets—short messages from Twitter, a major social media platform. Davidson et al. [11] have also used this modeling method (i.e., Logistic Regression) for tweet classification, but with *word level features*, *part-of-speech*, *sentiment*, and some *meta-data* associated with the tweets. *User features* (e.g., number of friends, followers, gender, geographic location, anonymity status, active vs. non-active status, among others) have also been shown to be useful in identifying aggressive and anti-social behavior [8, 9, 38, 44, 47]. These feature-based classifiers have several shortcomings. First, Fehn Unsvåg and Gambäck [12] have shown that user features only slightly improve the classifier’s ability to detect hate speech when tested on three Twitter data sets with a Logistic Regression model. Second, user features are often unavailable or not uniformly available across all social platforms.

Neural network models. In recent years, the research community has shifted away from traditional feature-based classification models toward neural network models, which we introduce briefly. *Recurrent Neural Networks* (RNN) [41] are a form of neural network which utilize an internal memory state to allow neural nodes to process input data of a variable length. *Convolutional Neural Networks* (CNN) [24] are models that rely on convolution, the mathematical operation to process slices of the input data rather than all of the input data at once by dividing the input data based on some notion of locality. *Long Short-Term Memory* (LSTM) [17] networks are an extension of RNNs that address the RNNs' inability to process long sequences by adding gate parameters that control how much impact the input data has on the internal memory state. *Transformers* [42] are a newer model class that uses the self-attention mechanism to replace recurrence and determine which parts of the input data are important for the subsequent layers.

Neural network models do away with the need for manual feature engineering and instead rely on the model itself to learn useful features. Badjatiya et al. [4] compared three approaches (CNN, LSTM, and FastText) for constructing contextualized word embeddings for potentially hateful words. Park and Fung [33] have used a neural network approach with two binary classifiers: a *Convolutional Neural Networks* (CNNs) with *word and character-level embeddings* for predicting abusive speech, and a Logistic Regression classifier with *n-gram features* for discriminating between different types of abusive speech (i.e., racism, sexism, or both). Zhang et al. [48] have applied pre-trained word embeddings and CNNs with *Gated Recurrent Units* (GRUs) to model long dependencies between features. Founta et al. [14] have built two neural classifiers: one for textual content and another one for user features. Their experiments concluded that joint training of the networks increases the overall performance. Pereira-Kohatsu et al. [36] presented a LSTM-MLP-based model dubbed *HaterNet* to hate speech in Spanish. Despite similarities in the model name (*HaterNet* vs our model HateNet), our work differs in its use of multiple datasets and transfer learning during training. More recently, transformer-based models using attention have become popular due to their strong performance across various natural language tasks. These models usually utilize large-scale pretraining on large text corpora and require fine-tuning to adapt the model toward a more specialized domain. Mozafari et al. [29] explored fine-tuning methods and successfully fine-tuned a BERT model pre-trained on English Wikipedia and BookCorpus for the task of hate speech detection. Awal et al. [2] applied multi-task learning to a BERT-based model with shared and private parameter layers, achieving strong performances.

Hate speech mapping, visualization and interpretation

Interpretation. An often-quoted shortcoming of neural network methods is their lack of interpretability, and research has been devoted to interpreting their decision-making and results. Park and Fung [33] have *clustered* the vocabulary of the Waseem [44] data set using the fine-tuned embedding from their model and found the clusters grouped sexist, racist, harassing etc. words. Wang [43] has presented the following three methods for interpretability: (1) *iterative partial occlusion* (i.e., masking input words) to study the network sensibility to the input length; (2) its opposite problem, called *lack of localization*, in which the model is insensitive to any region of the input; and (3) *maximum activations* of the final max-pooling layer of a

CNN-GRU network to identify the lexical units that contribute to the classification. According to their results, long inputs are more challenging to classify, and not all the maximum activated units are hateful.

Visualization. There have been several attempts at visualizing the hatefulness of speech with the aid of machine learning classification models. Capozzi et al. [7] employed an SVM model on an Italian hate speech Twitter dataset to generate word occurrence and co-occurrence visualizations, as well as Choropleth and Dorling maps to visualize the locations of where hateful tweets originate geographically. Modha et al. [28] explored several traditional and neural network classifiers for hate speech classification as either "Non-Aggressive", "Covertly Aggressive", or "Overtly Aggressive". They created a browser plugin that uses the final output of the models to color text based on the model classification; it displays a confidence score for each of the 3 classes and gives the option to automatically hide text that was identified as aggressive. Pereira-Kohatsu et al. [36] visualized the term frequency, user frequency, mention network, and term embedding of their LSTM-MLP model using T-SNE. Our work differs in our model architecture and we focus on the visualization of the sentence embeddings over terms.

Our work contributes the Map of Hate, a visualization that allows us to determine the type of speech employed (racist, sexist, etc.) and diagnose labeling errors.

Domain adaptation

Frustratingly Easy Domain Adaptation. Karan and Šnajder [21] have applied the *frustratingly easy domain adaptation* (FEDA) framework [10] to hate speech detection. This method works by joining two data sets from different domains in which their features are copied three times, depending on their presence in one or both data sets. The study concluded that domain adaptation boosts the classification performance significantly in six out of the tested nine cases (or data sets).

Multi-task learning. Multi-task learning addresses two or more tasks simultaneously while sharing knowledge between the tasks in order to improve performance on each of them. The core idea is that learning multiple similar tasks acts as a regularizer for the joint model, which is less prone to over-fitting and more generalizable. Waseem et al. [46] use domain adaptation in a multi-task learning framework to jointly learn from different data sets. They show that this can mitigate over-fitting in a hate speech detection context, a common problem when training models with small data sets. They do not utilize pre-trained embeddings in their work. Kapil and Ekbal [20] develop four different neural network models trained using a multi-task learning setup. They were able to achieve strong classification performance across all their experiments compared to baselines and state-of-the-art models. They differ from our work in model architecture as they utilize both shared weights and task-specific parameters. Rajamanickam et al. [39] developed two separate models: a double encoder based on LSTMs and a hard parameter sharing LSTM model, similar to our work. Our work differs from the above as we construct one prediction head per dataset, which improves prediction performances. While multi-task learning can be successfully applied to detect abusive language, care must be taken during training to make sure that one task does not dominate the other tasks. Also, it does not produce features and models that can be easily applied to other problems.

Transfer learning involves using features, weights, or any knowledge acquired for one task to solve another related problem. Transfer learning has been extensively used for domain adaptation and building models to solve problems where only limited data is available [31]. Formally, transfer learning involves the concepts of domains and learning tasks. Given a *source domain* D_S and *source learning task* T_S, a *target domain* D_T and *target learning task* T_T, transfer learning aims to make a contribution (i.e., improvement) to the learning of the *target predictive function* $f_T(\cdot)$ in D_T using the knowledge in D_S and T_S where $D_S \neq D_T$, or $T_S \neq T_T$ [31]. In this work, the label distribution of the two tasks is different, but related, since the two used data sets are annotated for analyzing different types of hate speech. $Y_S \neq Y_T$ where Y are the classes to be learned for the two tasks, respectively.

Transfer learning in hate speech detection has been previously explored. Pamungkas and Patti [30] examined the possibility of transfer learning across hate speech datasets of different languages using an LSTM-based model. They found that there was some ability for the transfer from other languages to English, but models were overall unable to match the performance of a model trained solely on datasets of the target language. The closest prior work to our own is that of Agrawal and Awekar [1], who applied transfer for cyber-bullying detection—another form of online abusive behavior. They experiment with transferring embeddings weights and network weights from models trained in a source domain to models fine-tuned in a target domain. Their transfer is one-directional (from source to target), which differs from our own approach ("Transfer learning setup") in which embedding weights are fine-tuned by solving both tasks at once: construct a single representation space and keep the classification tasks specific to each data set. Using our method, both tasks profit (similar to multi-task learning) and we generate embedding usable for future tasks.

Model

In this section, we first present our proposed deep neural network architecture, which inputs the raw text of tweets and predicts its hate category ("The hate speech detection pipeline"). Next, we augment the architecture to allow to jointly train and make predictions on multiple data sets ("Transfer learning setup").

The hate speech detection pipeline

Figure 1 highlighted in gray shows the conceptual schema of our model *HateNet*, which chains the five units detailed below.

The pre-processing unit. Compared to textual data gathered from other sources, Twitter data sets tend to be noisier. They contain misspellings, non-standard

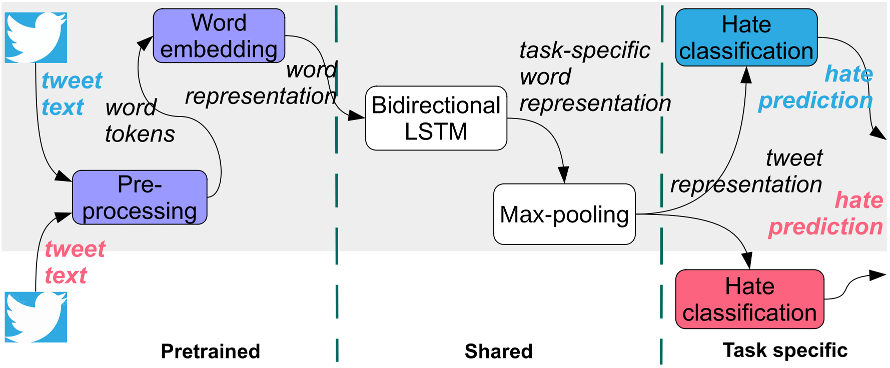


Fig. 1 The conceptual schema of HateNet (on gray background) and t-HateNet (whole schema)—our transfer learning architectures. (*HateNet*) The text of tweets is processed through two pre-trained units (shown in blue background), and three units whose parameters are trainable end-to-end via back-propagation. The output of the chain is the hate prediction for the input text. (*t-HateNet*) The tweets in two data sets (the red and the blue data set) are processed through a shared pipeline (pre-processing, ELMo, bi-LSTM and max-pooling) and a task-specific component (the Hate classification). The *tweet representation* space constructed at the output of the max-pooling unit is adequate for both learning tasks

abbreviations, Internet-related symbols, slang, and other irregularities. We pre-process and tokenize the textual data before training our learners. We remove repetitive punctuation, redundant white spaces, emojis, as well as *Uniform Resource Locators* (URLs). We add one space before every remaining punctuation. Note that we do not apply stemming nor remove stopwords. This results in a cleaner text than the original input.

The word embedding unit. In order to process natural language text, we map each English term to a numerical vectorial representation—dubbed the *word representation*. Training word representations typically requires amounts of textual data larger than our available hate speech data sets. Therefore, we opt to start from pre-trained word embedding models, among which we select the ELMo [37], as it is less resource-intensive to run compared with alternate technologies such as transformers. ELMo is itself a *Neural Network* model, which takes as its input a sentence, and outputs a vector representation for each word in the sentence. Peters et al. [37] show that ELMo's predictive performances are improved when used in conjunction with another (pre-trained) word embedding model. Here, we use the *Global Vectors* (GloVe) embedding [35] pre-trained on a data set of two billion tweets. ELMo constructs a lookup table between the words observed in the training set and their pre-trained representations.

When encountering a new word not present in the lookup table, ELMo falls back to constructing a character-level encoding, starting from its spelling. Polysemy—i.e., the fact that a word may have multiple possible meanings—is another problem for word embedding methods that employ lookup tables, as each word can only have one entry in the table and precisely one representation. ELMo addresses the issue of possibly multiple (and hence polysemous) vectors by first "reading" through the whole sentence and then *tweaking* the word representation according to the context;

the same word may have different representations in different sentences. In this work, we employ the pre-trained ELMo 5.5B [37] together with the Twitter-trained GloVe embeddings [35].

The bi-LSTM. The ELMo model is pre-trained for general purposes, and consequently, its constructed embeddings may have limited usefulness for the hate speech detection application. Thus, we add a bi-LSTM layer with randomly initialized weights to adapt the ELMo representation to the hate speech detection domain. The bi-LSTM module scans each sentence twice: from left to right and from right to left. This scanning produces two task-specific word representations for each word in the sentence—i.e., one from each scan.

The max-pooling unit. The next unit in the HateNet pipeline is the max-pooling layer; it constructs the embedding of an entire sentence starting from word representations. It inputs the two numerical representations of each word from the forward and backward pass of the bi-LSTM; it constructs a fixed-length vector by taking the maximum value for each dimension of the word representations. This results in a sentence representation defined in the same high-dimensional space as the word embeddings. Based on the output of the max-pooling unit, we can highlight which words contribute most to the classifier decision by counting how many times a word's dimension is selected for the sentence representation. This can be used as a proxy [19] to isolate the hateful content in a sentence (further discussed in "Main findings").

The hate classification unit. The last unit of the pipeline is a differentiable classifier consisting of a fully connected layer with a softmax activation layer. It inputs the previously constructed sentence representation and outputs the final prediction, which is used to calculate the Cross-Entropy Loss according to the ground truth and train the model. The weights of all the trainable modules in Fig. 1 (i.e., the Bi-LSTM, the Max-pooling, and the Hate classifier) are trained end-to-end via back-propagation. The following section gives the implementation details of each module.

Transfer learning setup

Here, we propose t-HateNet, an extension of HateNet to predict the type of hate speech in two unrelated data sets. The two models share the same base architecture, with the difference being that t-HateNet utilizes transfer learning to solve both tasks jointly. Intuitively, this allows insights learned from one task to be transferred to another, improving performance.

A mix of shared and individual processing units. Figure 1 shows *t-HateNet* the transfer learning schema that we have developed for leveraging multiple data sets and for solving multiple learning problems. Visibly, HateNet is an integral sub-part of t-HateNet. The pre-processing and the word embedding units are the same as the non-transfer settings, both having pre-trained weights. The bi-LSTM and the max-pooling units are also shared (i.e., they are processing text from multiple data sets and trainable via back-propagation). After we obtain the numeric representation for each tweet, we separate the data from different data sets and feed it into identically configured classifiers dedicated to each task. We make the final predictions for each data set independently from each other, as the prediction classes may be different. The entire processing pipeline is shared among all learning tasks apart from the final classification units. We learn a single word representation and a single sentence representation specific to all the tasks. Consequently, we leverage multiple specific data sets to train a larger, more general model for hate speech prediction.

Deep learners implementation details. Our proposed methods are implemented using the PyTorch library [34]. We use GloVe with 200 dimensions to initialize the word representations used by ELMo. The ELMo embeddings have 4,096 dimensions. We use a 2-layer stacked bi-LSTM with a hidden vector size of 512 dimensions. The final fully connected layer has a size of 128. We minimize the Cross-Entropy Loss during training using the Adam optimizer with the weight decay of 0.001 and an initial learning rate of 0.001. We train the model for 10 epochs using a batch size of 350. At every epoch, we test the model using a validation set, and the final weights are selected using the performance on the validation set. We tune the most critical learning parameters (i.e., the hidden state vector size in the bi-LSTM unit and the batch size of the learning process) via Grid Search on the validation set (see more details in the Online Appendix).

Datasets and experimental setup

The Waseem *data set* [44] is publicly available, and consists of 15,216 instances from Twitter annotated as *Racist*, *Sexist*, or *Harmless*.[1] This set is very imbalanced, with the majority class being *Harmless*, which is meant to reflect a real-world scenario where hate speech is less frequent than neutral tweets. The data selection appears biased, with most sexist tweets about a single television show in Australia and most racist tweets being Islamophobic. There are some questions about the quality of labeling in this data set, namely the number of false positives, considering that the data set was compiled to quantify the agreement between expert and amateur raters. Waseem [44] acknowledges this and observes that "the main cause of the error are false positives". Here are some examples of such *sexism* false positives:

- `@FarOutAkhtar How can I promote gender equality without sounding preachy or being a feminazi"? #AskFarhan`
- `i got called a feminazi today, it's been a good day.`
- `Yes except the study @Liberal_fem (the Artist Formerly known as Mich_something) offered's author says it does NOT prove bias @TamedInsanity`

[1] Note that both dataset denote the *Harmless* class as *Neither*. However, *Neither* has a limited meaning that is contextualized to a given dataset; we show in "Main findings" that the tweets in these classes overlap in our hate representation. We therefore interpret this class as non-hateful, i.e. *Harmless*.

- `In light of the monster derailment that is #BlameOneNotAll here are some mood capturing pics for my feminist pals.`

The Davidson *data set* [11] is also publicly available. It consists of 22,304 instances from Twitter annotated as *Hate*, *Offensive*, and *Harmless*. This data set was compiled by searching for tweets using the lexicon from Hatebase.org.

Hate speech prediction setup. We predict the types of hate speech (*hate*, *offensive* and *harmless* for Davidson, and *racism*, *sexist* and *harmless* for Waseem) using four classifiers: the Davidson and the Waseem baselines, and our approaches HateNet and t-HateNet. Given the class imbalance in both data sets, we over-sample the smaller classes to obtain a balanced data set. We have tried under-sampling the larger classes, but we obtained worse results. Unless otherwise stated, we train each classifier on 80% of each data set. We use 10% for validation and the remainder of 10% for testing. However, when studying the prediction with minimal amounts of data, we train on as little as 10% of the data set (see "Main findings"). We repeat the training and testing 10 times, reporting the mean and standard deviation. t-HateNet is trained on 80% of both data sets simultaneously; the other baselines are trained on each data set individually. We evaluate the prediction performance using the F1 measure—i.e., the geometric mean of precision and recall. A classifier must simultaneously obtain high precision and recall to achieve a high F1. We use the *macro-F1* so that smaller classes are equally represented in the final score.

Baselines. We compare our proposed method with two baselines: Waseem and Hovy [45] and Davidson et al. [11]. Waseem and Hovy [45] pre-processes the text by removing punctuation, excess white spaces, URLs, and stopwords and applying lower-casing and Porter stemmer for removing morphological and inflexional endings from words in English. They use a Logistic Regression model with character *n*-grams of lengths up to four as features. Davidson et al. [11] applies the same pre-processing as Waseem and Hovy [45] and also uses a Logistic Regression model. They included several word level features as 1–3 word *n*-grams weighted with TF-IDF and 1–3 Part-of-Speech *n*-grams. They also use two types of tweet-level features. The first type is readability scores, taken from a modified version of Flesch–Kincaid Grade Level and Flesch Reading Ease scores (with the number of sentences fixed to one). The second type is sentiment scores derived from the VADER sentiment lexicon design for social media [18]. In addition, they include binary and count indicator features for hashtags, mentions, retweets, URLs, and the number of words, characters, and syllables in each tweet. As they did not carry out an ablation study, it is unclear which features are most predictive for hate speech.

Main findings

Map of Hate Visualization. To understand the impact of the different modeling choices, we construct the *Map of Hate*—a two-dimensional visualization of the space of hateful text built using t-SNE [26], a technique originally designed to visualize high-dimensional data. Given the tweet representation built by HateNet

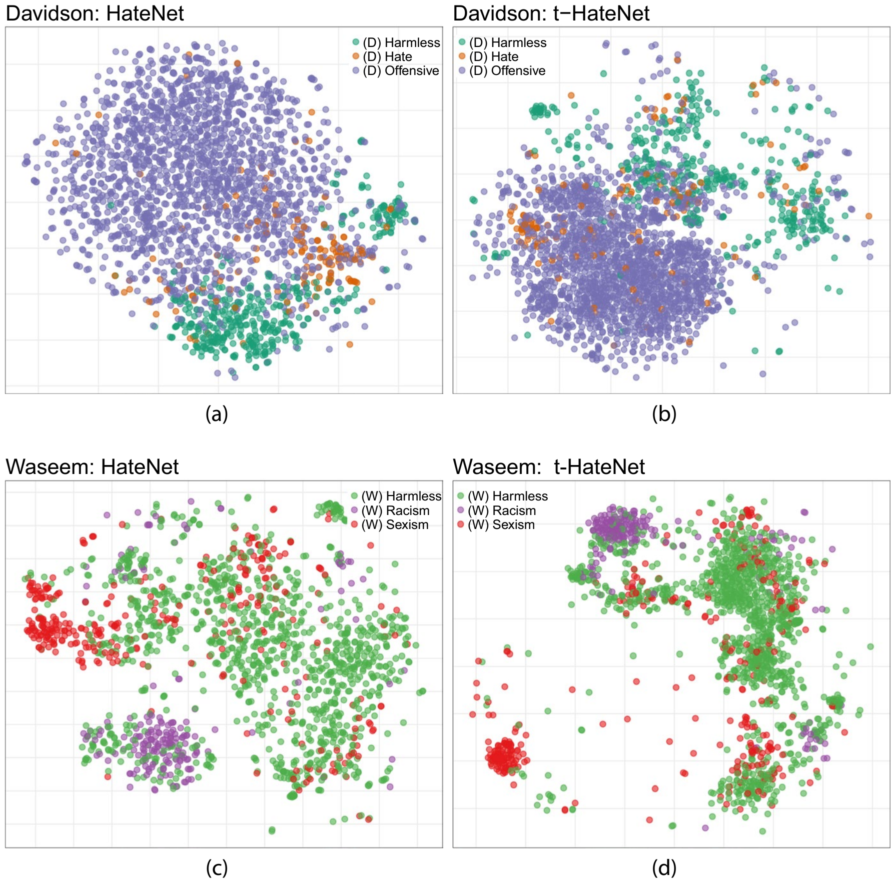
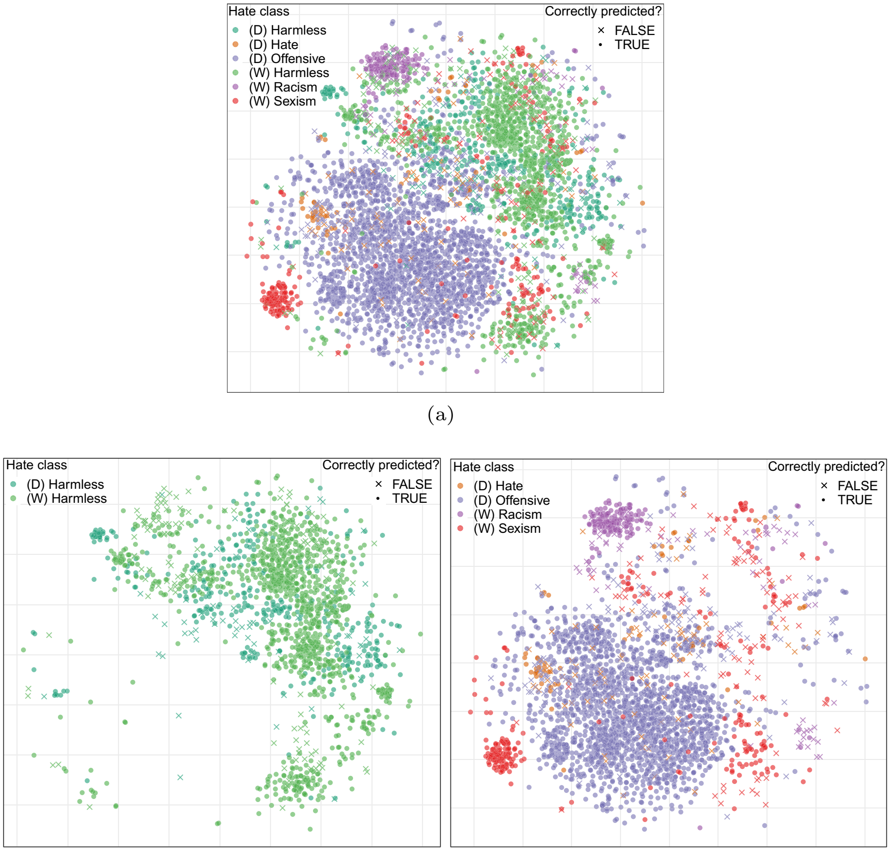


Fig. 2 The *Map of Hate* constructed on the Davidson data set (**a**, **b**) and the Waseem data set (**c**, **d**), using the tweet embeddings generated by HateNet (**a**, **c**) and t-HateNet (**b**, **d**). Note that the axes of the Map of Hate are synthetic and not interpretable

and t-HateNet (the output of the max-pooling units in Fig. 1), t-SNE builds a mapping to a 2D space in which the Euclidean pairwise distances correspond to the distance between pairs of tweet representation in the high-dimensional space. Figure 2a and c visualize the Map of Hate constructed by HateNet on a sample of 10% of the tweets Davidson and Waseem data sets, respectively. We observe that the tweets of different classes in Davidson appear clustered more closely together than in Waseem. Noticeably, the *racist* and *sexist* tweets in Waseem appear scattered throughout the *harmless* tweets. We posit this to be indicative of the false positives mentioned by Waseem [44].



(a)

(b)

(c)

Fig. 3 The *Map of Hate* constructed by t-HateNet jointly on 10% of the Davidson and Waseem data sets. Crosses and circles show incorrectly and correctly predicted examples, respectively. (Interactive version available online at https://bit.ly/39OVBgX). **a** All the six classes in the two data sets. **b** The Harmless classes of the two data sets overlap. **c** The hateful classes: Hate and Offensive (Davidson), Racism and Sexism (Waseem)

The Map of Hate has several usages. First, it quickly maps each type of hateful content into two-dimensional space regions and explains why a tweet is predicted as hateful. This can allow us to uncover patterns in the dataset, such as particular language and labeling errors. For example, all the tweets in the red bottom-left cluster in Fig. 2d start with “I’m not sexist, but...” (many wrongly labeled as sexist, see next). Second, using the interactive version of the map[2] we can help distinguish between the failed and successful cases—there is a large number of failed predictions in the sexist (red) tweets (Fig. 3c) sprinkled in the top-right area of the harmless tweets (observed in Fig. 3b). Interestingly, this might be linked to implicit

[2] Interactive Map of Hate available online: https://bit.ly/39OVBgX.

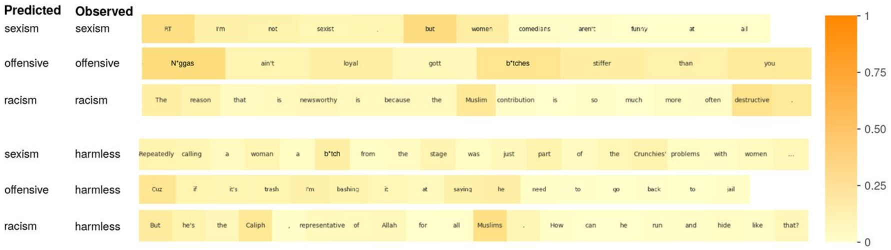


Fig. 4 *Warning: this figure contains real-world examples of offensive language!* Automatically highlighting offensive terms. It is not possible to illustrate the performance of the technique without quoting actual examples of speech classified as hate speech and to that end we include this figure. Examples of six tweets from the Davidson and Waseem data sets, together with their predicted and observed hate category. The top three tweets are hateful (*sexist*, *offensive* and *racist* respectively) and their category was correctly predicted. The bottom three tweets are *harmless*, but they were incorrectly predicted as hateful. The color map shows how many times a word's representation was selected for the tweet representation, normalized by the size of the embedding (here 512)

vs. explicit gender biases; we know that people with implicit gender bias tend to use phrases like "I am not sexist" to begin their sentences [27] (Fig. 4). Finally, it can visualize the impact of constructing task-specific tweet embeddings (see the Online Appendix).

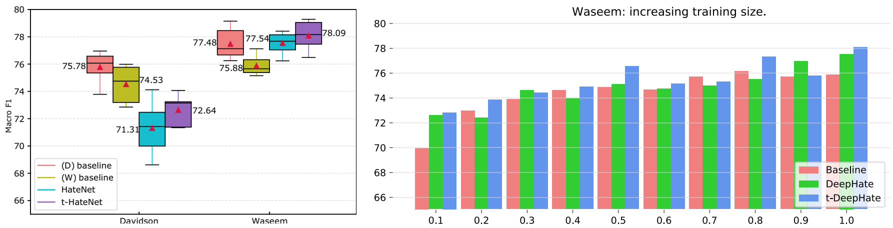
The effects of transfer learning. Figure 2b and d show the tweet sample in Davidson and Waseem data sets, respectively, projected in the joint space constructed by t-HateNet. The Davidson tweets appear to lose their clustering and break down into subgroups, especially for the harmless class. We can interpret this lack of clustering as the class is less distinct in the space, which causes difficulties for the classification head to distinguish them from other classes. This explains the slightly lower prediction results obtained by t-HateNet on Davidson. The situation is reversed on Waseem, where tweets belonging to different hate categories are clustered more tightly together—explaining the better performances on this data set. The decrease in clustering indicates that the quality of representation on Davidson is slightly negatively impacted by labeling quality in Waseem. However, the prediction on Waseem benefits from a large increase—the new space is more adequate to represent its tweets thanks to the Davidson data set.

Figure 3a visualizes the tweets from both Davidson and Waseem projected into the t-HateNet space in the same figure. The circles mark correctly predicted tweets by t-HateNet while the crosses mark the incorrectly predicted tweets (best seen in the interactive version). Figure 3b further details only the *harmless* tweets from both data sets, and Fig. 3c the hateful tweets from both data sets (*racism* and *sexism* from Waseem, and *hate* and *offensive* from Davidson). Several conclusions emerge. First, the hateful and harmless content appears separated in the joint space (bottom-left for hateful and top-right for harmless). Second, the harmless tweets from both data sets appear overlapped (Fig. 3b), which is correct and intuitive since both classes stand for the same type of content. Third, the hateful content (Fig. 3c) has a more complex dynamic: the *offensive* (Davidson) tweets occupy most of the space, while most of the *sexist* and *racist* (Waseem) appear tightly clustered on the sides. This is indicative of the sampling bias in Waseem. Lastly, the *sexist* and *racist* (Waseem) sprinkled tweets throughout the *harmless* are appear overwhelmingly miss-classified, which can be explained by the false positives in Waseem.

Highlighting hateful content. Here we describe a method to highlight hateful content in a text. The max-pooling is a dimension-wise max operation, i.e., for each dimension, it selects the maximum value over all word embeddings in the forward and backward pass of the Bi-LSTM. Assume a total of n words in the sentence, each represented by a numerical vector with d dimensions. For each of the d dimensions, the max-pooling picks the maximum value across the n words. We count the number of times each word is selected to represent the whole sentence across the d dimensions, and we normalize the scores by d (a word can be picked a maximum of d times). This constructs a score between zero (a word is never selected) and 1 (the word is always selected). The higher the word's score, the more representative the word's representation for the final sentence representation and the hate prediction. Despite the recent critiques about linking attention mechanisms to word importance [19], we find that this method succeeds in highlighting hateful content and can be used to identify patterns in hatefully labeled content, as shown next.

Correctly and incorrectly predicted examples. Figure 4 shows six examples of tweets in our data sets, with each word highlighted according to its score. (*Warning! this figure contains real-world examples of offensive language*). The top three examples are correctly predicted as *sexist*, *offensive* and *racist*, respectively. We observe that most tweets labeled as *sexism* (Waseem data set) start with "I'm not a sexist, but ..." and variations. While this might raise questions about data sampling bias in the data set construction, this behavior is captured in the higher weights assigned with "but" (ending of "I'm not a sexist, but"), "woman" and (strangely) "RT" (i.e., retweet). In the *offensive* example (Davidson data set), we notice that offensive slang words are correctly scored higher. We also notice that *racism* examples (such as the third tweet in Fig. 4) tend to refer exclusively to the Islamic religion and its followers—which can bias the learned embedding. However, for this example, we observe that words such as "destructive" are correctly recognized as indicators of hate speech. The bottom three lines in Fig. 4 show examples where *harmless* tweets are incorrectly labeled as hateful. This happens mainly due to their writing style and choice of words. The tweet misclassified as *sexism* uses language similar to sexism to draw awareness against it, and as a result, it is classified as sexist. Similarly, the incorrectly labeled *offensive* example is written in a style similar to other offensive tweets in our data set. Finally, the falsely *racist* tweet uses Islam-related terminology, and because of the data sampling bias, it is classified as racist.

Predicting hate speech in two data sets. We measure the prediction performances of HateNet and t-HateNet against the two baselines, making several observations based on the prediction performances for each classifier and each data set (Fig. 5a). First, our models HateNet and t-HateNet outperform the baselines on the Waseem data set, and they under-perform them on the Davidson data set. We posit this is due to the baselines' external information—statistics, user information, and tweet meta-data. Upon investigating the weights associated



(a) (b)

Fig. 5 **a** Prediction performances on two data sets: Davidson and Waseem. Boxplot summarizing macro-F1 score for each data set, and each approach (baselines, HateNet, t-HateNet). Red diamonds and values indicate mean F1. Each box consists of 10 independent runs. **b** Prediction performances with limited amounts of training data on Waseem datasets. The *x*-axis shows the percentage of the training set used for training, the *y*-axis shows the macro-F1 measure. Each bar shows the mean value over 10 runs, and the standard deviation

with each feature by the logistic regression, we observe that among the 161 features with a non-negligible absolute weight ($> 10^{-5}$), 21 features are non-textual and Twitter-related. The tweet sentiment (quantified using VADER [18]) is the most relevant, with a high negative weight for *hate* and *offensive* classes, and with a high positive class for the *harmless* class. Important tweet features include the number of words, user mentions, hashtags, characters, and syllables of a tweet. We chose not to use this additional information in our approaches, as it would render the obtained results applicable solely to the Twitter data source. Furthermore, t-HateNet confuses the *Hate* class with *Offensive* in 41% of the cases (more details are included in the Online Appendix). By manually inspecting some failed predictions, we notice that it is challenging even for humans to differentiate between purposely hateful language (with a particular target in mind) and generally offensive texts (without a target).

Second, we observe that the Davidson baseline outperforms the Waseem baseline on both Davidson and Waseem data sets. This shows that the external features built by Davidson are more informative for Twitter-originating hate speech than Waseem's. Third, we observe that t-HateNet outperforms HateNet on both data sets, admittedly not by a large margin.

Predicting hate speech using limited amounts of data. The advantage of jointly leveraging multiple data sets emerges when only limited amounts of labeled data are available. Here, we study the typical situation in which it is required to learn a hate speech classifier with minimal amounts of labeled data but with the help of a larger unrelated hate speech data set. We restrain the amount of training data: after sampling the training set (80% of the data set), the validation set (10%), and the testing set (10%), we further subsample the initial training set so that only a percentage is available for the model training. For t-HateNet, we perform this additional subsampling for only one of the data sets, keeping all training examples of the other data set. We vary the percentage of training data in the downsampled data set, and Fig. 5a shows the mean prediction performance and its standard deviation on the Waseem data set (Davidson is shown in the Online Appendix). The performances of HateNet and t-HateNet are positively correlated with the size of the training set. Visibly, we can observe that t-HateNet generally outperforms HateNet across all but two of the ten experiments, showcasing transfer learning's ability to alleviate the issue of limited datasets and improve classification performance.

Conclusion

With our social interactions and information being increasingly online, more and more emphasis is placed on identifying and resolving Internet issues in our society. It is important to make social media safer by detecting and reducing hateful, offensive, or otherwise unwanted social interactions.

In this paper, we introduced HateNet and t-HateNet, two machine learning and natural language processing pipelines for differentiating harmless tweets from racist, sexist, hateful, or offensive messages on Twitter. HateNet and t-HateNet share the same architecture (therefore, the name similarity); t-HateNet leverages a transfer learning procedure to transfer knowledge from one task to another by constructing a shared generalized representation of hate from both datasets. We empirically show that this improves classifier detection and helps to address data scarcity when there is a limited amount of available data.

We further constructed the Map of Hate, a two-dimensional visualization of the generalized embedding space of hate. We showcase several use cases for the Map of Hate in identifying patterns in hateful speech, identifying errors in labeling in the datasets, identifying cases where the model struggles in its prediction, and in examining the similarity of the datasets through examining the separation of classes between the two datasets in the embedding space.

Our methods contribute to analyzing social media contents at scale to make these web platforms safer and better understand the genre of hate speech and its sub-genres. Our automated text processing and visualization methods can separate different types of hate speech and explain what makes text harmful. Their use could even reduce the need to expose human moderators and annotators to distressing messaging on social media platforms.

Limitations and future works

Here we discuss the two main limitations of this work, their likely causes and possible future steps to alleviate them.

Limited performance gain. The first limitation relates to our transfer learning architecture's somewhat modest performance improvement, as showcased by our experiments. We note, however, that the improvement is consistent across our experiments. We argue this is linked to the amount of information available to the model—the two datasets not being fully transferable due to differences in the labeling criteria. There are two main approaches for increasing the quantity of information available for learning. The traditional Machine Learning approach requires increasing the number of labeled examples (i.e., the number of rows in the training dataset). This requires significant manual effort and reinforces bias and subjectivity (see next paragraph). Our work proposes an alternative: increasing the number of smaller, unrelated data sets to learn jointly and to transfer information between apparently unrelated learning tasks. This provides the required additional information and reduces the overall bias. The increase in the number of datasets can be relatively easily achieved as various resources provide hate speech datasets to be added to our framework.[3]

Learning human-annotated biases. The second limitation concerns Machine Learning models' reliance on bias-rich, human-annotated data. This limitation applies to all supervised machine learning classification algorithms and is outside the scope of this work. It is worth noting and discussing some of its consequences for its downstream applications in social science. We start from the observations that hate speech does not have a universally agreed upon formal definition [15]. Furthermore, what individuals perceive as hateful is context-, political-environment, and individual-life-course-dependent. As a result, human-generated labels of hate speech are subjective and embed biases and preconceptions. As machine learning models (like the ones we build in this work) learn from examples, they will also learn these inherent biases and further reproduce them in automatic labeling. Understanding and ideally limiting biases is essential for social science applications before deploying machine-labeled data in downstream analysis. Without a precise and operationalizable definition of hate speech, we may be unable to circumvent human annotations for model training.

However, our proposed approach partially alleviates the problem. Learning using multiple datasets can significantly reduce bias and subjectivity in hate speech detection models. When we increase the size of the training set by coding more data, as is typical in Machine Learning (see previous paragraph), it is crucial to consider the limitations of relying on the same limited set of labelers. Doing so risks reinforcing their biases and subjectivity as they label the additional data. This narrow focus on a limited group of labelers and a specific labeling criterion restricts the model's ability to detect different facets of hate speech. For example, if the dataset primarily focuses on sexism, the trained model may struggle to identify hate speech targeting immigrants or other marginalized groups. In contrast, training the model with multiple datasets exposes it to a larger pool of labelers and a diverse range of hate speech facets. Each dataset brings a different perspective and definition of hate speech, encompassing various societal biases. The model learns from this wider range of interpretations, effectively reducing the impact of individual biases and enabling a more comprehensive understanding of hate speech.

In summary, leveraging multiple smaller, unrelated data sets to learn jointly and transfer information between apparently unrelated learning tasks can help with performance, bias and subjectivity issues. Future work will explore leveraging

[3] The website https://hatespeechdata.com/ offers a collection of hate speech datasets from multiple works, making them highly accessible for integration.

a more significant number of small datasets and analyze the impact on prediction subjectivity.

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/s42001-023-00224-9.

Acknowledgements This research was supported by an Australian Government Research Training Program (RTP) Scholarship, by an ANU Research School of Social Sciences Cross-College Grant, by the Commonwealth of Australia (represented by the Defence Science and Technology Group) through a Defence Science Partnerships Agreement, and by the Australian Department of Home Affairs. This research was undertaken with the assistance of resources and services from the National Computational Infrastructure (NCI), which is supported by the Australian Government.

Funding Open Access funding enabled and organized by CAUL and its Member Institutions.

Data availability statement The datasets used in this study are available from the corresponding author on reasonable request.

Declarations

Conflict of interest On behalf of all authors, the corresponding author states that there is no conflict of interest.

References

1. Agrawal, S., & Awekar, A. (2018). Deep learning for detecting cyberbullying across multiple social media platforms. In G. Pasi, B. Piwowarski, L. Azzopardi, A. Hanbury (Eds.) *Advances in information retrieval*. ECIR 2018. Lecture notes in computer science, vol 10772. Cham: Springer. https://doi.org/10.1007/978-3-319-76941-7_11
2. Awal, M. R., Cao, R., Lee, R. KW., & Mitrović, S. (2021). AngryBERT: Joint learning target and emotion for hate speech detection. In: K. Karlapalem et al. (Eds.) *Advances in knowledge discovery and data mining*. PAKDD 2021. Lecture Notes in Computer Science, vol 12712. Cham: Springer. https://doi.org/10.1007/978-3-030-75762-5_55
3. Badjatiya, P., Gupta, M., & Varma, V. (2019). Stereotypical bias removal for hate speech detection task using knowledge-based generalizations. In *WWW'19* (pp. 49–59). ACM Press.
4. Badjatiya, P., Gupta, S., Gupta, M., & Varma, V. (2017). Deep learning for hate speech detection in tweets. In *WWW '17* (pp. 759–760). ACM Press.
5. Barreto, M., Ellemers, N., Cihangir, S., & Stroebe, K. (2009). *The self-fulfilling effects of contemporary sexism: How the well-being and behavior of women is affected by the subtle discrimination they encounter* (pp. 99–124). American Psychological Association.
6. Cambridge dictionary: hate speech.
7. Capozzi, A., Lai, M., Basile, V., Poletto, F., Sanguinetti, M., Bosco, C., Patti, V., Ruffo, G., Musto, C., Polignano, M., Semeraro, G., & Stranisci, M. (2020). "contro l'odio'': A platform for detecting,

monitoring and visualizing hate speech against immigrants in Italian social media. *Italian Journal of Computational Linguistics, 6*, 77–97.

8. Chatzakou, D., Kourtellis, N., Blackburn, J., De Cristofaro, E., Stringhini, G., & Vakali, A. (2017). Mean birds: Detecting aggression and bullying on Twitter. In *WebSci '17* (pp. 13–22). ACM Press.
9. Cheng, J., Danescu-Niculescu-Mizil, C., & Leskovec, J. (2015). Antisocial behavior in online discussion communities. In *ICWSM'15* (pp. 61–70). AAAI Press.
10. Daume III, H. (2007). Frustratingly easy domain adaptation. In *Proceedings of the 45th annual meeting of the association of computational linguistics, Prague, Czech Republic* (pp. 256–263). Association for Computational Linguistics.
11. Davidson, T., Warmsley, D., Macy, M. W., & Weber, I. (2017). Automated hate speech detection and the problem of offensive language. In *ICWSM'17*. AAAI Press.
12. Fehn Unsvåg, E., & Gambäck, B. (2018). The effects of user features on Twitter hate speech detection. In *Workshop on abusive language online (ALW2)* (pp. 75–85). Association for Computational Linguistics.
13. Fortuna, P., & Nunes, S. (2018). A survey on automatic detection of hate speech in text. *ACM Computing Surveys, 51*(4), 1–30.
14. Founta, A. M., Chatzakou, D., Kourtellis, N., Blackburn, J., Vakali, A., & Leontiadis, I. (2019). A unified deep learning architecture for abuse detection. In *Proceedings of the 10th ACM conference on web science (WebSci '19)* (pp. 105–114). New York, NY, USA: Association for Computing Machinery. https://doi.org/10.1145/3292522.3326028
15. Hietanen, M., & Eddebo, J. (2023). Towards a definition of hate speech—with a focus on online contexts. *Journal of Communication Inquiry, 47*(4), 440–458. https://doi.org/10.1177/01968599221124309
16. Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation, 9*, 1735–80.
17. Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation, 9*(8), 1735–1780.
18. Hutto, C. J., & Gilbert, E. (2014). Vader: A parsimonious rule-based model for sentiment analysis of social media text. In *ICWSM'14*. AAAI Press.
19. Jain, S., & Wallace, B. C. (2019). Attention is not explanation. In *North American chapter of the Association for Computational Linguistics* (pp. 3543–3556). Association for Computational Linguistics.
20. Kapil, P., & Ekbal, A. (2020). A deep neural network based multi-task learning approach to hate speech detection. *Knowledge-Based Systems, 210*, 106458.
21. Karan, M., & Šnajder, J. (2018). Cross-domain detection of abusive language online. In *Proceedings of the 2nd workshop on abusive language online (ALW2)* (pp. 132–137). Brussels, Belgium: Association for Computational Linguistics.
22. Kumar, R., Ojha, A. K., Malmasi, S., & Zampieri, M. (2018). Benchmarking aggression identification in social media. In *Proceedings of the first workshop on trolling, aggression and cyberbullying (TRAC-2018), Santa Fe, New Mexico, USA* (pp. 1–11). Association for Computational Linguistics.
23. Lange, J., Mollas, I., & Tsoumakas, G. (2018). Hatebusters: A web application for actively reporting YouTube hate speech. In *Proceedings of the 27th international joint conference on artificial intelligence* (pp. 5796–5798). International Joint Conferences on Artificial Intelligence.
24. LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature, 521*(7553), 436.
25. Liu, H., Burnap, P., Alorainy, W., & Williams, M. L. (2019). Fuzzy multi-task learning for hate speech type identification. In *The World Wide Web conference on—WWW '19* (pp. 3006–3012). ACM Press.
26. van der Maaten, L., & Hinton, G. (2008). Visualizing data using t-SNE. *JMLR, 9*(Nov), 2579–2605.
27. Merritt, A. C., Effron, D. A., & Monin, B. (2010). Moral self-licensing: When being good frees us to be bad. *Social and Personality Psychology Compass, 4*(5), 344–357.
28. Modha, S., Majumder, P., Mandl, T., & Mandalia, C. (2020). Detecting and visualizing hate speech in social media: A cyber watchdog for surveillance. *Expert Systems with Applications, 161*, 113725.
29. Mozafari, M., Farahbakhsh, R., & Crespi, N. (2020). A BERT-based transfer learning approach for hate speech detection in online social media. In *Complex networks and their applications VIII* (pp. 928–940). Cham: Springer.
30. Pamungkas, E. W., & Patti, V. (2019). Cross-domain and cross-lingual abusive language detection: A hybrid approach with deep learning and a multilingual lexicon. In *Proceedings of the 57th annual*

meeting of the association for computational linguistics: Student research workshop, Florence, Italy (pp. 363–370). Association for Computational Linguistics.

31. Pan, S. J., & Yang, Q. (2010). A survey on transfer learning. *Transactions on Knowledge and Data Engineering, 22*(10), 1345–1359.
32. Paradies, Y., Ben, J., Denson, N., Elias, A., Priest, N., Pieterse, A., Gupta, A., Kelaher, M., & Gee, G. (2015). Racism as a determinant of health: A systematic review and meta-analysis. *PLoS One, 10*, 1-45.
33. Park, J. H., & Fung, P. (2017). One-step and two-step classification for abusive language detection on Twitter. In *Workshop on abusive language online* (pp. 41–45). Association for Computational Linguistics.
34. Paszke, A., Gross, S., & Chintala, S., et al. (2017). Automatic differentiation in pytorch. In *NeurIPS'17*.
35. Pennington, J., Socher, R., & Manning, C. (2014). Glove: Global vectors for word representation. In *EMNLP'14* (pp. 1532–1543). Association for Computational Linguistics.
36. Pereira-Kohatsu, J. C., Quijano-Sánchez, L., Liberatore, F., & Camacho-Collados, M. (2019). Detecting and monitoring hate speech in twitter. *Sensors, 19*(21).
37. Peters, M. E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., & Zettlemoyer, L. (2018). Deep contextualized word representations. In *Proceedings of NAACL* (pp. 2227–2237). Association for Computational Linguistics.
38. Pitsilis, G. K., Ramampiaro, H., & Langseth, H. (2018). Effective hate-speech detection in Twitter data using recurrent neural networks. *Applied Intelligence, 48*(12), 4730–4742.
39. Rajamanickam, S., Mishra, P., Yannakoudakis, H., & Shutova, E. (2020). Joint modelling of emotion and abusive language detection. In D. Jurafsky, J. Chai, N. Schluter, & J. Tetreault (Eds.), *Proceedings of the 58th annual meeting of the association for computational linguistics* (pp. 4270–4279). Association for Computational Linguistics (ACL). https://doi.org/10.18653/v1/2020.acl-main.394
40. Reuters (2018). Why Facebook is losing the war on hate speech in Myanmar. https://www.reuters.com/investigates/special-report/myanmar-facebook-hate/. Accessed 16 May 2019.
41. Rumelhart, D. E. & McClelland, J. L. (1987). Learning internal representations by Error Propagation. In *Parallel distributed processing: Explorations in the microstructure of cognition: Foundations* (pp. 318–362), MIT Press.
42. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L. U., & Polosukhin, I. (2017). Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, & R. Garnett (Eds.), *Advances in neural information processing systems.* (Vol. 30). Curran Associates Inc.
43. Wang, C. (2018). Interpreting neural network hate speech classifiers. In *Workshop on abusive language online (ALW2)* (pp. 86–92). Brussels, Belgium: Association for Computational Linguistics.
44. Waseem, Z. (2016). Are you a racist or am I seeing things? Annotator influence on hate speech detection on Twitter. In *Workshop on NLP and computational social science* (pp. 138–142). Austin, TX, USA: Association for Computational Linguistics.
45. Waseem, Z., & Hovy, D. (2016). Hateful symbols or hateful people? Predictive features for hate speech detection on twitter. In *NAACL student research workshop* (pp. 88–93). Association for Computational Linguistics.
46. Waseem, Z., Thorne, J., & Bingel, J. (2018). Bridging the gaps: Multi task learning for domain transfer of hate speech detection. *Online Harassment* (pp. 29–55). Cham: Springer. https://doi.org/10.1007/978-3-319-78583-7_3
47. Wulczyn, E., Thain, N., & Dixon, L. (2017). Ex machina: Personal attacks seen at scale. In *World Wide Web, WWW '17* (pp. 1391–1399). ACM Press.
48. Zhang, Z., Robinson, D., & Tepper, J. (2018). Detecting hate speech on Twitter using a convolution-GRU based deep neural network. In A. Gangemi, R. Navigli, M. Vidal, P. Hitzler, R. Troncy, L. Hollink, A. Tordai, M. Alam (Eds.), *ESWC 2018: The semantic web* (pp. 745–760).

Authors and Affiliations

Lanqin Yuan[1] · **Tianyu Wang**[2] · **Gabriela Ferraro**[2] · **Hanna Suominen**[2,3] · **Marian-Andrei Rizoiu**[1,2]

✉ Lanqin Yuan
lanqin.yuan@student.uts.edu.au

Tianyu Wang
tianyu.wang2@anu.edu.au

Gabriela Ferraro
gabriela.ferraro@anu.edu.au

Hanna Suominen
hanna.suominen@anu.edu.au

Marian-Andrei Rizoiu
marian-andrei.rizoiu@uts.edu.au

[1] University of Technology Sydney, Sydney, NSW, Australia

[2] The Australian National University, Canberra, ACT, Australia

[3] University of Turku (UTU), Turku, Southwest Finland, Finland