Hate Speech Detection is Not as Easy as You May Think: A Closer Look at Model Validation

Aymé Arango
aarango@dcc.uchile.cl
Department of Computer Science
University of Chile
IMFD, Chile

Jorge Pérez
jperez@dcc.uchile.cl
Department of Computer Science
University of Chile
IMFD, Chile

Barbara Poblete
bpoblete@dcc.uchile.cl
Department of Computer Science
University of Chile
IMFD, Chile

ABSTRACT

Hate speech is an important problem that is seriously affecting the dynamics and usefulness of online social communities. Large scale social platforms are currently investing important resources into automatically detecting and classifying hateful content, without much success. On the other hand, the results reported by state-of-the-art systems indicate that supervised approaches achieve almost perfect performance but only within specific datasets. In this work, we analyze this apparent contradiction between existing literature and actual applications. We study closely the experimental methodology used in prior work and their generalizability to other datasets. Our findings evidence methodological issues, as well as an important dataset bias. As a consequence, performance claims of the current state-of-the-art have become significantly overestimated. The problems that we have found are mostly related to data overfitting and sampling issues. We discuss the implications for current research and re-conduct experiments to give a more accurate picture of the current state-of-the art methods.

CCS CONCEPTS

• **Computing methodologies** → **Machine learning approaches**; *Cross-validation*; • **Information systems** → *Social tagging*.

KEYWORDS

hate speech classification, experimental evaluation, social media, deep learning

ACM Reference Format:
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 (SIGIR '19), July 21–25, 2019, Paris, France.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3331184.3331262

SIGIR '19, July 21–25, 2019, Paris, France

ACM ISBN 978-1-4503-6172-9/19/07…$15.00
https://doi.org/10.1145/3331184.3331262

1 INTRODUCTION

Automatic detection of hate speech has become an increasingly relevant research topic in the past few years [11, 26, 27]. The worldwide adoption of online social networks has created an explosion in the volume of text-based social exchanges. Social media communications can strongly influence public opinion and some social platforms are said to have enough social capital to influence the outcome of democratic processes [10]. Therefore, correctly assessing hate speech and other forms of online harassment has become a pressing need, to guarantee non-discriminatory access to digital forums, among other things [9].

Large social media providers, such as Facebook and Twitter have mechanisms for users to report hate speech. However, this approach requires efficient automatization techniques for the evaluation of such content, which does not appear to be simple: user accounts that constantly post potentially dangerous hateful expressions have incorrectly been deemed as harmless, and blatantly offensive content can go unreported for long periods of time [20]. Given the enormous volume of content posted daily in these platforms, human editorial approaches have become unfeasible. Hence, the incorrect assessment of toxic content can be most likely attributed to the lack of reliable mechanisms for its automatic detection. Twitter, for example, has publicly declared its commitment to *"serve healthy conversations"* and *"to help increase the collective health, openness, and civility of public conversation, and to hold ourselves publicly accountable towards progress."*[1]. Among other things, Twitter has even announced funding initiatives for academic research on this topic.[2]

Despite the apparent difficulty of the hate speech detection problem evidenced by social-media providers, current state-of-the-art approaches reported in the literature show near-perfect performance. Within-dataset experiments on labeled hate-speech datasets using supervised learning achieve F1 scores above 93% [1, 2, 6, 11]. Nevertheless, there are only a few studies towards determining how generalizable the resulting models are, beyond the data collection upon which they were built on, nor on the factors that may affect this property [18]. Furthermore, recent literature that surveys current work also views the state of the art under a more conservative and cautious light [11, 18].

In this work, we take a close look at the experimental methodology utilized for achieving the results described by the state-of-the-art methods. We focus on two methods reporting the best results for hate-speech detection over Twitter data: the work by Badjatiya

[1] https://twitter.com/jack/status/969234275420655616
[2] https://blog.twitter.com/official/en_us/topics/company/2018/measuring_healthy_conversation.html

et al. [2] (93% F1 – *WWW* 2017), and by Agrawal and Awekar [1] (94% F1 micro and macro-average F1 – *ECIR* 2018). At first, our intention was to replicate these findings to then measure how these models would perform on similar yet different datasets. However, a closer look at the papers and the code provided by the authors for replicating experiments, revealed details in their implementation which can produce data overfitting. In both cases there were very subtle issues that are not directly apparent from the description of the methods nor from the companion code. For the case of the work by Badjatiya et al. [2], the issue is produced in the way that the authors compute features from the input data. In the work by Agrawal and Awekar [1] the issue is produced by how the authors perform the oversampling of the minority classes.

To study the effects of the aforementioned methodological issues that we observed in prior work, we replicated these methods exactly as presented by the authors. This was done to ensure we could obtain their reported performance using their code and the same data. Next, we made corrections to avoid data-overfitting and re-evaluated the generalization error of such approaches.

As a result, we were able to give a more accurate picture of the actual performance of current state-of-the art methods (at least for the particular datasets in which they are trained and tested). Our re-evaluation indicated a significant drop in the main metrics. In particular, the performance of both methods measured in F1 dropped from over 90% (Table 1) to below 80% (Table 2). In addition, we found that this performance dropped even further to below 50% F1 (Table 3) when the resulting models were applied to different datasets (even if these were from the same domain).

We were able to attribute the poor cross-dataset generalization of these models to overfitting due to bias in the benchmark dataset. In particular, one of the most widely used datasets, that of Waseem and Hovy [30], which contains thousands of labelled hate-speech messages, has a strong user bias. Most of the hateful messages were generated by only a few users: This is, 65% of the messages marked as hateful ("sexist" or "racist") in Waseem and Hovy's dataset [30] were produced by only 2 users. Since this bias is not evident from the dataset description, prior works using this data [1, 2, 6, 13, 18, 23, 32] have trained on the same sets of users on which they have later evaluated their models, involuntarily inducing a *user-overfitting* effect. To prove this, we empirically show that when previously unseen users are sampled for testing, the performance of state-of-the-art models drops below 44% macro-average F1 for the same dataset (Table 4). We finally try to alleviate this issue by enriching the Waseem and Hovy [30] dataset with a different yet similar dataset [8] restricting the maximum number of tweets per user. We show that the generalization of the methods increases when the user distribution is more carefully taken into account (Tables 5 and 6).

In summary, our work shows that although state-of-the-art methods report impressive performances [1, 2], hate-speech detection is far from solved. We provide an explanation for this by exposing some methodological issues, but also by showing the impact of some inherent biases in the datasets that are publicly available and widely used [30]. In light of our findings we believe that it is important to pay careful attention to experimental evaluation and how predictive models generalize. In addition, it becomes urgent to disclose more information about existing datasets, in particular, information about user distribution, to allow researchers to improve their sampling techniques.

Most of the code and datasets that we used are already available online and we mention each source when describing the specific datasets and methods. Nevertheless, for reproducibility issues, we made available a centralized repository including all the code and datasets[3] used in this paper.

2 RELATED WORK

The field of hate-speech automatic detection and classification has evolved rapidly in the past years. Interest has increased as social media and social platforms have grown in terms of influence and user adoption. Similarly to the field of *sentiment analysis*, automatic hate speech identification research has stemmed from two types of approaches: those based on the use of lexicons and those based on machine learning. In this literature overview we focus on machine learning approaches for detecting hate speech in social media textual content. Most work focuses on hate speech detection for Twitter messages, also known as *tweets*, which are short text messages. Thus, we also focus our review on works that use this type of data.

Several previous works have tried a diverse range of *classic* machine-learning strategies [5, 8, 19, 30]. These usually have an initial feature-extraction phase, such as computing Term-Frequency Inverse-Document-Frequency scores or Bag-of-Words vectors, but also combining it with meta-information such as information from the user account and information about the network structure (followers, replies, etc.) [5, 22, 28]. These features are then used as input for methods such as Logistic Regression, SVM, or Random Forest classifiers. Surveys on the topic [11, 26, 27] provide a nice general view on these methods for Twitter and other domains.

More recently, Deep Learning methods have attracted interest to hate-speech detection [1, 2, 13, 18, 23, 32]. As opposed to more traditional methods, Deep-Learning methods are able to automatically learn representations of the input data that can be used as features to classify it [15]. One distinctive characteristic of Deep Learning approaches is that they solve the problems *end-to-end* using the labelled data for feature extraction and classification at the same time [15]. Architectures such as Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), in particular Long Short-Term Memory networks (LSTMs) and Gated Recurrent Units (GRUs), are nowadays the methods of choice for several Natural Language Processing tasks [14]. In the context of hate speech, Gambäck and Sikdar [13] and Park and Fung [23] use CNNs for classification achieving near 80% F1 over the Waseem and Hovy's dataset [30]. Similar results are reported by Zhang et al. [32] using an architecture that combines CNNs and GRUs. The best results are by Badjatiya et al. [2] and Agrawal and Awekar [1] reporting over 93% F1. Both methods are based on LSTMs (see Section 4 for a detailed exposition of these methods).

There is some recent work on testing the generalization of the state-of-the-art methods to other datasets and domains [1, 6, 18]. Most of this work has been focused on Deep Learning methods. Agrawal and Awekar [1] test the performance of models trained

[3] https://github.com/aymeam/User_distribution_experiments

on tweets [30] classifying on Wikpedia data [31] and Formspring data [25]. The authors show that transfer learning from Twitter to the two other domains performs poorly achieving less than 10% F1. In a similar study, Dadvar and Eckert [6] perform transfer learning from Twitter to a dataset of Youtube comments [7] showing a performance of 15% F1. Gröndahl et al. [18] present a comprehensive study reproducing several state-of-the-art models. Specially important for us is the experiment transferring Badjatiya et al.'s model [2] trained on the Waseem and Hovy's dataset [30] to two other similarly labeled tweet datasets [8, 32]. Even in this case the performance drops significantly, obtaining 33% and 47% F1 in those sets. This is a 40+% drop from the 93% F1 reported by Badjatiya et al. [2]. From these results,Gröndahl et al. [18] draw as a conclusion that model architecture is less important than the type of data and labeling criteria being used. In this paper our results are coherent with those of Gröndahl et al. [18]. However, we take our research a step further by investigating why this issue occurs.

3 DATASETS

One of the most important aspects for supervised learning is the availability of labeled data. Therefore, we deem it necessary to pay close attention to how the data used for hate speech analysis is composed. We note that, due to the sensitive matter of hate-speech content most of the existing datasets contain only text and labels, but purposefully omit message and user identifiers. However, as we argue in Section 6, for the particular task of hate-speech detection, the knowledge of whether a certain user generated a set of messages can be of critical importance. With the tweet identifiers one is able to, for example, have access to the user distribution on the hate vs non-hate texts. Next, we detail the datasets that we use in our current work along with relevant properties obtained from our data exploration process (some properties were not evident from the original dataset papers).

Waseem and Hovy [30]. In most of our reported experiments we use the public annotated dataset constructed by Waseem and Hovy [30]. Although there is no standard dataset for hate-speech detection, Waseem and Hovy's dataset is one of the most popular ones[4]. It is used in several hate-speech detection studies [1, 2, 6, 13, 18, 23, 32] and it is consistently mentioned as an important source in the surveys about the subject [11, 26, 27]. This dataset is composed of 16K tweet identifiers[5] annotated as "sexist", "racist" and "non-hate". We recovered the tweets using the Twitter API.[6] At the time of recovering (late 2018), some of the tweets had been eliminated and some users had been suspended. Therefore, the dataset we obtained was composed of 14,949 tweets, of which 4,839 are hateful tweets (racist or sexist) and the rest (10,110) are labelled as non-hateful.

As we have mentioned (and as we show in Section 6), knowing the user distribution is a decisive feature when training classifiers. For the Waseem and Hovy's dataset only 1,590 users generates all the data, from which 491 generate all the "sexist" tweets, and only 8 users generate all the "racist" ones. Moreover, just a few users generate almost all the hateful data. For the "sexist" label, there is a single user that generates 40% of all tweets, and for "racist" label, there is a single user that generates more that 90% of all tweets.

Davidson et al. [8]. Davidson et al.'s dataset [8] was constructed combining an automatic search for tweets containing words in a hate-speech lexicon and an annotation using CrowdFlower workers. The resulting set is composed by 24,802 tweets annotated in three classes: *hate speech*, *offensive but not hate speech*, or *neither offensive nor hate speech*. Although the original version of this dataset contains only the text of each tweet and its label, the authors were kind enough to provide us with additional information in order for us to identify which tweets are produced by the same users.[7]

SemEval dataset [4]. We consider the SemEval 2019 dataset for the "Multilingual detection of hate speech against immigrants and women in Twitter" task [4]. This dataset is composed of 9,000 tweets, 3,783 of which are labelled as hateful and 5,217 as not hateful. The publicly available version of this dataset only provides message text and labels, no message identifiers nor user identifiers were provided. We note that as we did for the Davidson et al. [8] dataset, we wrote to the owners of the dataset requesting any information that could help know which messages were produced by a same user. However, in this exchange the SemEval team declined our request on account of user privacy concerns under GDPR,[8] but more importantly, on the basis that hate-speech detection should be focused on language and not users.[9]

In our current work, we do not argue against user privacy preservation measures nor compliance under federal regulations. Nevertheless, we argue in favor of the scientific value of knowing if a certain hate speech message has been generated by the same source. In this sense, we show empirical evidence in Section 6 that language is not necessarily the only focus that research on this topic should have.

4 REPLICATING THE STATE OF THE ART

In this section we take a closer look at the methodologies for hate speech detection introduced by Badjatiya et al. [2] and by Agrawal and Awekar [1]. According to our literature review and surveys on the topic, discussed in Section 2, these approaches constitute the state of the art for which we can attempt reproducibility, in the sense that they provide code and use available datasets. Overall, we provide a detailed description of the experimental methodologies used by the authors of these works. In particular, we include many details that are not explicit in the original papers for these methods. We obtained these details after careful analysis of the source codes provided as companion material for the original publications. Hence, as a means of performing a sanity check, in this section we include reproducibility experiments using the same datasets as in prior works, obtaining almost identical results to those of the authors. The purpose of this analysis is to discuss in Section 5 the methodological issues that we identify from our study of these approaches and their implications.

[4] 189 citations according to Google Scholar as of January 2019.
[5] http://github.com/zeerakw/hatespeech
[6] https://developer.twitter.com/en/docs/api-reference-index.html
[7] This data was obtained only for research purposes through personal communication with the authors.
[8] General Data Protection Regulation https://eugdpr.org/
[9] https://groups.google.com/forum/#!topic/semeval2019-task5-hateval/wIxx6jWlJng

We note that both of the aforementioned works use Deep Neural Network architectures, hence this section assumes certain background knowledge since it is not possible for us to cover many details on this topic at this point. Nevertheless, for more information Goodfellow et al.'s book [15] provides a detailed description of each of the mentioned architectures. We also refer the reader to the excellent survey by Goldberg [14] that introduced several of the needed concepts specifically tailored for Natural Language Processing.

4.1 Badjatiya et al. [2], WWW 2017

Badjatiya et al. [2] experimented with various traditional machine-learning models, deep neural models and the combination of both. The best results that they reported was a 93% F1.[10] This result was obtained by using a combination of a Recurrent Neural Network to construct *word embeddings*, and a decision-tree method that uses these embeddings to classify a piece of text. A *word embedding* is a dense-vector representation (vector of real values) of a word or token [16, 21, 24]. The main idea is that semantically-similar words are assigned to vectors that are close to each other in the vector space [21]. Word embeddings are a crucial piece in the current performance of Deep Learning methods for Natural Language Processing [14]. Although word embeddings are usually learned in an unsupervised way from big corpora, one can also learn embeddings specifically tailored for some task if labelled data is available. This last strategy is the one used by Badjatiya et al. [2]. They divided their method in two phases:

Phase 1. The authors first use an architecture composed of an Embedding Layer (dimension=200), followed by a Long Short-Term Memory (LSTM) network (dimension=50), and finally a fully connected layer with 3 neurons plus a Softmax activation to produce the probabilities for classes "sexist", "racist" and "non-hate". This architecture is trained *end-to-end* using the labelled tweets with cross-entropy as loss function and the Adam optimizer with the default learning rate for 10 epochs. Between the layers they use Dropout (probabilities 0.25 and 0.5, respectively) for regularization. Although this architecture can be directly used for the task of predicting the class of an input text, the authors use it only as a *feature extractor* for Phase 2 (i.e., the embeddings learned in the Embedding Layer are used as a feature space for the next phase).

Phase 2. Consider an input tweet t as the sequence of words $t = (w_1, w_2, \ldots, w_k)$. The authors used the Embedding Layer trained in the previous phase to produce a sequence of vectors $E_t = (\mathbf{x}_{w_1}, \mathbf{x}_{w_2}, \ldots, \mathbf{x}_{w_k})$, and then averaged them to obtain a single vector $\overline{E}_t$. They then, use a Gradient-Boosted [12] Decision Tree (GBDT) that has as input the vectors $\overline{E}_t$ for every tweet t, and train it using the labels for every tweet.

We were able to reproduce Badjatiya et al.'s reported performance (see Table 1), following closely their paper description [2] and the companion code[11]. We first use the whole dataset to train the embeddings in Phase 1. Then we train Phase 2 with a 10-fold cross validation. In Table 1 we report precision, recall and F1 for every class, and also a micro and macro average for all classes. We obtained a micro-average F1 score of 94%, and a macro-average F1 of 93%. The results are consistent with the results reported by the authors [2]. We note that in this particular setting the embeddings were obtained using the complete labeled dataset. We discuss this aspect of the methodology in more detail in Section 5, since it induces overfitting, as features selection must be performed after separating training and testing data.

Table 1: Replication of results of the state of the art [1, 2].

Method	Class	Prec.	Rec.	F1
Badjatiya et al. [2]	Neither	95.5	96.8	96.1
Emb. over all dataset	Racist	94.5	93.5	94.0
	Sexist	91.2	87.5	89.3
	Micro avg.	94.6	94.6	94.6
	Macro avg.	93.7	92.6	93.1
Agrawal and Awekar [1]	Neither	95.1	91.7	93.4
Oversamp. all dataset	Racist	94.9	96.0	95.4
	Sexist	92.5	97.0	94.6
	Micro avg.	94.4	94.4	94.4
	Macro avg.	94.2	94.9	94.5

4.2 Agrawal and Awekar [1], ECIR 2018

Agrawal and Awekar [1] also experimented with several models. Similar to Badjatiya et al. [2] they observed that deep neural models outperformed other methods. They reported results for Convolutional networks, LSTMs, Bidirectional LSTMs (BiLSTM), and BiLSTMs with Attention [3]. For all architectures the results reported were similar [1], thus we only reproduce the results for BiLSTMs. The detailed architecture is almost exactly the same as Phase 1 in Badjatiya et al. [2] with the sequence of layers Embedding → Recurrent → FullyConnected → Softmax. The recurrent layer is a BiLSTM, in particular, a set of two LSTMs layers that process the input in both directions concatenating the outputs of both LSTMs at each step. The authors reported their results for embeddings of dimension 50, and both LSTMs of dimension 50 (effective dimension 100). They also used Dropout with the same probabilities as [2] and trained their model with the Adam optimizer for 10 epochs [1].

One of the keypoints of Agrawal and Awekar [1] was the impact of oversampling. As the authors observed, training datasets for hate-speech detection contain only a few posts marked as "hate", and they claimed that the class imbalance problem can be tackled by oversampling the class with fewest examples [1]. They started their process by oversampling the entire dataset, even before beginning with the cross validation process. This is counter intuitive, since oversampling must be done only on the training data partitions and not on the test data, otherwise the model will be overfitted to the artificial bootstrapping result (we detail this in Section 5). Nevertheless, we replicated their exact same process and we were able to reproduce their reported results with 94% micro and macro-average F1 score (see Table 1).

[10] In [2] the authors did not mention if the F1 reported is a micro or macro average. We include both metrics in our experiments.

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

5 THREATS TO THE VALIDITY OF THE STATE OF THE ART

In this section we discuss threats to the validity of the results reported in prior works, which we presented in Section 4. First, in Sections 5.1 and 5.2 we detail the methodological issues that we were able to find and how they affect the validity of the reported results. Specifically, we show that by modifying the methodology in each case –to avoid data overfitting problems– and then re-conducting experiments, produces model performance to drop significantly in both cases. Secondly, in Section 5.3 we study the validity of the results reported in prior work when applying these models to different datasets from the same domain.

Deep Learning methods are sensible to hyperparameter tunning such as changing the optimization algorithm, its learning rate, or the dimensions of layers [15]. Nevetheless in the experiments in this section we just use the methods proposed by Badjatiya et al. [2] and Agrawal and Awekar [1] with the same hyperparameters presented in their papers and companion code, without doing any additional hyperparameter search. We left as future work a detailed hyperparameter tunning to see how it further affects the performance of the methods.

5.1 Extracting features using the entire dataset

As discussed in Section 4.1, the method of Badjatiya et al. [2] was divided in two: feature extraction (Phase 1) and classification (Phase 2). This is a typical pipeline for a supervised-learning problem. However, the methodological issue in this case is that the features in Phase 1 are obtained by considering the complete labeled dataset. More specifically, let $\mathcal{T}$ be the labelled dataset and assume that it is divided as $\mathcal{T} = \mathcal{T}_{\text{train}} \cup \mathcal{T}_{\text{test}}$. For the feature extraction, the whole set $\mathcal{T}_{\text{train}} \cup \mathcal{T}_{\text{test}}$ is used, while for the classification phase, the set $\mathcal{T}_{\text{train}}$ is used to train the classifier, and $\mathcal{T}_{\text{test}}$ is used to evaluate it. Thus, when one sees the complete pipeline, the set $\mathcal{T}_{\text{test}}$ is simultaneously used to train the complete architecture and also validate it. This may lead to an artificial increase in the performance of the model; we are underestimating the generalization error due to overfitting.

To empirically support our claim above, we re-run the same method proposed by [2] but this time extracting features only from the set $\mathcal{T}_{\text{train}}$ (by using the LSTM-based architecture), then training the GBDT classifier with these features over the same set $\mathcal{T}_{\text{train}}$, and reporting all the metrics over $\mathcal{T}_{\text{test}}$. Table 2 shows the results for a 10-fold cross validation reporting precision, recall, and F1 for the three classes. By only making this change, we can observe that the metrics obtained are significantly lower than those reported in the original paper. This is, F1 for each class decreases from 96.1 to 88.1 (Neither), 94.0 to 70.2 (Racist), and 89.3 to 60.9 (Sexist), which implies a macro-average F1 drop of 20 points (from 93.1 to 73.1).

5.2 Oversampling before the train-test split

The main issue that we were able to identify in the experimental design by Agrawal and Awekar [1], discussed in Section 4, is the oversampling phase. As the authors describe in their paper they oversampled the data from "hate" class thrice, that is, they replicated "hate" posts 3 times in their original dataset. As their results show, this oversampling led them to an increase from ~ 70 F1 to more than 90 F1. Taking a closer look at the companion code provided by the authors, the pipeline for the oversample process is as follows: (1) first they oversample the complete dataset, then they (2) split the dataset into train-test, and finally (3) train the model (see function `run_model` in the `DNNs.ipynb` file in the companion code[12]).

Table 2: Replication of the results of the state-of-the-art taking into account the methodological issues mentioned in Section 5

Method	Class	Prec.	Rec.	F1
Badjatiya et al. [2]	Neither	82.3	94.7	88.1
Emb. over train set	Racist	78.0	64.0	70.2
	Sexist	84.5	47.8	60.9
	Micro avg.	82.3	82.1	80.7
	Macro avg.	81.6	68.9	73.1
Agrawal and Awekar [1]	Neither	90.3	86.5	88.3
Oversamp. train set	Racist	69.6	81.3	75.0
	Sexist	74.0	77.4	75.5
	Micro avg.	84.7	84.1	84.3
	Macro avg.	78.0	81.7	79.6

Methodologically, oversampling the data before the train-test split (i.e., before cross-validation) produces model overfitting to the resulting data. This is why oversampling should be done after train-test data partitions are generated. To see this with numbers, assume that we have an initial unbalanced dataset $\mathcal{T}$ and we first triplicate every tweet with a "hate" label to obtain a new dataset $\mathcal{T}'$ where the portion of "hate" and "non-hate" labels is balanced (50%-50%). For simplicity, assume that in the obtained dataset there are no other repeated tweets beside the ones that we replicated. Now let us partition $\mathcal{T}'$ with a 15%-85% split into $\mathcal{T}'_{\text{test}} \cup \mathcal{T}'_{\text{train}}$. One can prove (see the appendix for a detailed exposition) that for every "hate"-labelled tweet t, the probability that t appears simultaneously in $\mathcal{T}'_{\text{test}}$ and $\mathcal{T}'_{\text{train}}$ is approximately 38%. With only this observation one can construct a trivial classifier for the tweets in the test set as follows: for every $t \in \mathcal{T}'_{\text{test}}$, if $t \in \mathcal{T}'_{\text{test}} \cap \mathcal{T}'_{\text{train}}$ then we classify t as "hate", and if $t \in \mathcal{T}'_{\text{test}} \setminus \mathcal{T}'_{\text{train}}$ then we classify t as "non-hate". Notice that we do not make any mistake when we classify a tweet as "hate", thus for the "hate" class we obtain 100% precision, but only 38% recall. This gives us an F1-score of 55%. On the other hand, since every "non-hate" tweet is classified correctly, then we obtain 100% recall for the "non-hate" class. Computing the precision for this class is a bit more complicated. First note that there is a 62% (100% − 38%) of "hate" tweets in the test set that we classify as "non-hate". Thus there is a total of 31% (0.62×0.5) of the whole test set that is missclassified as "non-hate" and a 50% that is correctly classified. This gives us a precision of 61% (0.5/(0.5 + 0.31)) for the "non-hate" class, and then we obtain an F1-score of 76%. In summary, we have 55% F1 for "hate" and 76% F1 for "non-hate" which gives us an average of 66% F1. This result, which is 16% over the trivial baseline, is obtained for free just by oversampling the data.

The net effect of the oversampling performed by [1] is actually very similar to the issue described in the previous section; there

[12] https://github.com/sweta20/Detecting-Cyberbullying-Across-SMPs/blob/master/DNNs.ipynb

Table 3: Results using the Waseem and Hovy's dataset [30] as training set and SemEval 2019 [4] as testing set

Method	Class	Prec.	Rec.	F1.
Badjatiya et al. [2] on SemEval	None	60.1	95.9	73.9
	Hateful	69.1	12.5	21.1
	Micro avg.	63.8	60.8	51.6
	Macro avg.	64.6	54.2	47.5
Agrawal and Awekar [1] on SemEval	None	59.7	93.5	72.9
	Hateful	59.5	13.2	21.6
	Micro avg.	59.6	59.7	59.7
	Macro avg.	59.6	53.3	47.2

is a portion of the test data that is also used to train the classifier which may lead to an artificial increase in the performance of the model due to overfitting to the test set.

To empirically support our claim we re-conducted the same method proposed by Agrawal and Awekar [1] but this time making first the train-test splitting and then oversampling the train set before training the models. Table 2 shows the results for a 10-fold cross validation of this model. The metrics are significantly lower than the ones reported in the original paper [1]. This is, F1 for each class decreases from 93.4 to 88.3 (Neither), 95.4 to 75.0 (Racist), and 94.6 to 75.5 (Sexist) which implies a macro-average F1 drop of 15 points (from 94.5 to 79.6).

5.3 Generalization to other datasets

Table 2 shows a more conservative picture of the actual performance of the state-of-the-art methods. A natural question at this point is how well do these models generalize to other dataset from the same domain. To estimate this, in our next experiment we evaluate those models –generated on the complete Waseem and Hovy dataset– on the SemEval 2019 dataset [4] (see Section 3).

The SemEval dataset considers only two classes: "hate" vs "non-hate". Thus, to evaluate the Badjatiya et al. [2] and Agrawal and Awekar [1], we consider these models as binary classifiers; whenever the methods classify a text as either "sexist" or "racist", we mark it as "hateful" (and as "None" otherwise). Table 3 shows the results of this experiment in which we can observe that model performance on SemEval is much lower, hence evidencing that the models do not generalize well to new data. More noticeably, the "hateful" class classification drops in F1 score from $\sim$ 70 to 21.1 for [2] and from $\sim$ 75 to 21.6 for Agrawal and Awekar [1].

6 THE IMPACT OF USER DISTRIBUTION ON MODEL GENERALIZATION

Similarly to what has been reported in fake news generation in social media [17], hate speech generation can be attributed to a very small portion of the overall community. This creates a considerable class imbalance problem, which if not taken explicitly into account can create important data bias issues in labeled datasets. In this sense, user distribution can have a considerable impact in the classification results. In particular, if the tweets in the datasets are generated by only a small number of different users, then one could potentially reduce the hate-speech detection problem to a user-identification problem.

For example, in the Waseem and Hovy's dataset [30], only 20% of the tweets has the "sexist" label, and 12% the "racist" label. Moreover, there is a single user that generates 44% of "sexist" comments, and a single user that generates 96% of "racist" comments. To illustrate the effect of user bias, let's call the 2 aforementioned users u_s and u_r, respectively. Then, as a *thought experiment* let's assume that there is a method that correctly identifies the tweets produced by these two users. Thus, one could create a trivial classifier that assigns label "sexist" to every tweet from u_s, assigns label "racist" to every tweet from u_r, and assigns "non-hate" label to every other tweet. One can prove that given the distribution of users plus the imbalance between the classes, this simple classifier that only identifies two users gives 87% micro and macro-average F1.

Hence, we hypothesize that users distribution is another source of overestimation in the performance of the state-of-the-art classifiers [1, 2]. In order to support our hypothesis, we conducted some additional experiments in which the data is split into train-test sets, such that no tweet from a same user is simultaneously in both sets. With this type of data sampling we ensure that identifying the author of a message is totally useless for the classification. In other words, we ensure that the classifier does not learn a secondary task (i.e., identifying users) instead of the task for which it is being trained for (i.e., identifying hate speech).

Table 4 shows the results over the Waseem and Hovy's original dataset under this new sampling requirement. We performed three different partitions of users in the original dataset, ensuring that no user is repeated between train and test set, and also ensuring at least an 85% of tweets of each class are in the train set. We report the average metrics for the three partitions for the binary classification problem (details on the partitions can be found in the appendix). However, given that the original dataset is extremely skewed, when forcing the classifier not to learn users by avoiding having the same users in training and testing, model performance drops dramatically (Table 4).

These results show that models trained on the Waseem and Hovy dataset [30], can be prone to severe *user overfitting* (i.e., the model learns to identify users). In light of these results, it is our belief that to avoid user overfitting issues we must resort to less biased datasets. This is datasets in which hate speech data has been sampled by a better distribution of users. Hence, the Waseem and Hovy dataset, which has been widely used, is not a good fit to solve this problem. In the following section, we present work towads improving this existing dataset.

6.1 Fixing the user-overfitting problem

We attempt to alleviate the *user-overfitting* issue by enriching the Waseem and Hovy's dataset [30] with the Davidson et al. dataset [8]. The whole idea is to limit the number of tweets of the most prolific users in the "hateful" class or less represented class in order to avoid overfitting the model to prolific users.

We re-sample the data in the Waseem and Hovy dataset by placing a limit of at most 250 tweets per class for each user. The resulting dataset was reduced to 5,576 tweets, 1,490 of which were in the "hateful" class. Since, even if using this criteria we still had

Table 4: Results using partitions of the Waseem and Hovy's dataset [30] into train set and test set considering the user distribution (no overlapping users between train and test sets)

Method	Class	Prec.	Rec.	F1
Badjatiya et al. [2]	None	49.6	93.4	64.3
	Hateful	68.8	15.4	23.5
	Micro avg.	63.8	54.1	46.1
	Macro avg.	59.2	54.4	43.9
Agrawal and Awekar [1]	None	47.5	98.0	63.0
	Hateful	75.3	03.5	06.7
	Micro avg.	62.3	48.4	35.1
	Macro avg.	61.4	50.8	34.9

Table 5: Results using a combination of the datasets of Waseem and Hovy [30] and Davidson et al. [8] with no overlapping users between train and test sets

Method	Class	Prec.	Rec.	F1
Badjatiya et al. [2]	None	83.3	92.4	87.6
	Hateful	79.5	61.4	69.3
	Micro avg.	82.1	82.3	81.7
	Macro avg.	81.4	76.9	78.4
Agrawal and Awekar [1]	None	85.8	73.2	77.4
	Hateful	70.2	80.3	74.2
	Micro avg.	80.0	78.1	77.7
	Macro avg.	78.0	76.7	75.8

the "hateful" class dominated by few users we enriched this existing dataset with additional users. To achieve this we added all of the hateful tweets from the Davidson et al. dataset (preserving the limit of at most 250 hateful messages per user), which resulted in an overall of 7,006 examples, 2,920 of which corresponded to the "hateful" class.[13]

In Table 5 we present the results for a 10-fold cross validation considering partitions with no overlapping users between the train and test sets using the enriched dataset. The performance of the model on this new dataset should provide better generalization to previously unseen data, with a performance across datasets more similar to that reported with cross-validation. To corroborate the generalization of the resulting model we use our newly created dataset to train the models proposed by Badjatiya et al. [2] and by Agrawal and Awekar [1]. Then we evaluate these models on previously unseen data by classifying tweets in the SemEval 2019 set. The results of this process are shown in Table 6. We can observe that in comparison to the results reported in Table 3 the metrics improved substantially, specially the F1 score in the "hateful" class. This is, when we used the original (unaltered) Waseem and Hovy dataset for training and the SemEval dataset for testing, generalization for the "hateful" class was poor, with F1 score for both models of just 21% (Table 3). Nevertheless, when training on our newly created dataset, the same metric improves to 52% F1 for the Badjatiya et al. method and improved to 62% for the Agrawal and Awekar method (Table 6). The macro average F1 also improved from 47% for both methods (Table 3) to 51% and 54% respectively (Table 6). We stress that this improvement is obtained even with less examples; the new dataset contains half of the examples of the Waseem and Hovy's original dataset (7,006 vs 14,949), but a better user distribution. This is additional evidence supporting our hypothesis that user distribution is an important factor when training for hate-speech detection.

[13] https://github.com/aymeam/User_distribution_experiments

Table 6: Results using a combination of the datasets of Waseem and Hovy [30] and Davidson et al. [8] as train set ensuring a maximum number of tweets per user, and the SemEval 2019 dataset [4] as test set

Method	Class	Prec.	Rec.	F1
Badjatiya et al. [2]	None	62.2	42.4	50.4
	Hateful	44.8	64.4	52.8
	Micro avg.	54.9	51.7	51.4
	Macro avg.	53.5	53.4	51.6
Agrawal and Awekar [1]	None	78.0	32.5	45.9
	Hateful	48.4	87.3	62.3
	Micro avg.	65.5	55.5	52.8
	Macro avg.	63.2	59.9	54.1

7 CONCLUDING REMARKS AND FUTURE WORK

Our work shows that there is an overestimation of the performance of current state-of-the-art approaches for hate-speech detection on twitter. In particular, we evidence potential methodological issues that induce data overfitting that affect the validity of the results reported by the works that we analyzed [1, 2]. In addition, we discuss a previously unknown issue, that of *user overfitting* induced by important user bias in labeled data. In this sense, we believe that hate speech detection is not only about language, but also about users.

Our findings contribute to understand why these methods that show impressive performance in their original test sets do not generalize well to other datasets, even if new data comes from the same domain. Both methods that we analyzed are based on Deep Learning techniques. Deep Learning models present two orthogonal issues that should be taken into consideration when using these types of approachess: they easily overfit the data, and they are difficult to explain [15]. This is a dangerous combination as small issues may mislead researchers into over-optimistic conclusions. As we show, extremely high performances should be better analyzed.

We also perform experiments designed to alleviate state-of-the-art issues and improve model generalization. Our proposed solutions show improved cross-dataset performance and better model performance estimation, providing a more accurate take on the state of the art.

We conclude that it is important to pay more detailed attention to experimental evaluation and how existing methods generalize. In particular, we believe that it is urgent to disclose more information

about user distribution in existing datasets. In this regard, we agree with the need to preserve user privacy. However, we believe that for the purpose of scientific investigation, at least there should be disclosure on whether there is an important portion of messages coming from the same users. Or even better, to produce datasets that do not contain important user bias.

For future work, one of our goals is to do hate-speech detection in a cross-lingual scenario. Compared to the English language, there is considerable less resources for other languages. Thus, transfer learning from one language to another is a line of research that we would like to explore. For this, we believe that improving generalization of existing methods for English is an important first step.

ACKNOWLEDGMENTS

We thank Thomas Davidson for providing all the information concerning the dataset described in Davidson et al. [8]. This work was supported by the Millennium Institute for Foundational Research on Data (IMFD). Poblete was also funded by Fondecyt grant 1191604.

REFERENCES

[1] Sweta Agrawal and Amit Awekar. 2018. Deep Learning for Detecting Cyberbullying Across Multiple Social Media Platforms. In *Advances in Information Retrieval - 40th European Conference on IR Research, ECIR 2018, Grenoble, France, March 26-29, 2018, Proceedings*. 141–153. https://doi.org/10.1007/978-3-319-76941-7_11

[2] Pinkesh Badjatiya, Shashank Gupta, Manish Gupta, and Vasudeva Varma. 2017. Deep learning for hate speech detection in tweets. In *Proceedings of the 26th International Conference on World Wide Web Companion*. International World Wide Web Conferences Steering Committee, 759–760.

[3] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural Machine Translation by Jointly Learning to Align and Translate. *CoRR* abs/1409.0473 (2014). http://arxiv.org/abs/1409.0473

[4] Valerio Basile, Cristina Bosco, Viviana Patti, Manuela Sanguinetti, Elisabetta Fersini, Debora Nozza, Francisco Rangel, and Paolo Rosso. [n.d.]. Shared Task on Multilingual Detection of Hate. SemEval 2019, Task 5, https://competitions.codalab.org/competitions/19935.

[5] Despoina Chatzakou, Nicolas Kourtellis, Jeremy Blackburn, Emiliano De Cristofaro, Gianluca Stringhini, and Athena Vakali. 2017. Mean Birds: Detecting Aggression and Bullying on Twitter. In *Proceedings of the 2017 ACM on Web Science Conference, WebSci 2017, Troy, NY, USA, June 25 - 28, 2017*. 13–22. https://doi.org/10.1145/3091478.3091487

[6] Maral Dadvar and Kai Eckert. 2018. Cyberbullying Detection in Social Networks Using Deep Learning Based Models; A Reproducibility Study. *CoRR* abs/1812.08046 (2018). arXiv:1812.08046 http://arxiv.org/abs/1812.08046

[7] Maral Dadvar, Dolf Trieschnigg, and Franciska de Jong. 2014. Experts and Machines against Bullies: A Hybrid Approach to Detect Cyberbullies. In *Advances in Artificial Intelligence - 27th Canadian Conference on Artificial Intelligence, Canadian AI 2014, Montréal, QC, Canada, May 6-9, 2014. Proceedings*. 275–281.

[8] Thomas Davidson, Dana Warmsley, Michael W. Macy, and Ingmar Weber. 2017. Automated Hate Speech Detection and the Problem of Offensive Language. In *Proceedings of the Eleventh International Conference on Web and Social Media, ICWSM 2017, Montréal, Québec, Canada, May 15-18, 2017*. AAAI Press, 512–515. https://aaai.org/ocs/index.php/ICWSM/ICWSM17/paper/view/15665

[9] Laure Delisle, Alfredo Kalaitzis, Krzysztof Majewski, Archy de Berker, Milena Marin, and Julien Cornebise. 2018. A large-scale crowd-sourced analysis of abuse against women journalists and politicians on Twitter. (2018).

[10] Nicholas Fandos and Kevin Roose. 2018. Facebook Identifies an Active Political Influence Campaign Using Fake Accounts. https://www.nytimes.com/2018/07/31/us/politics/facebook-political-campaign-midterms.html. [Online; accessed 26-January-2019].

[11] Paula Fortuna and Sérgio Nunes. 2018. A Survey on Automatic Detection of Hate Speech in Text. *ACM Comput. Surv.* 51, 4 (2018), 85:1–85:30.

[12] Jerome H. Friedman. 2000. Greedy Function Approximation: A Gradient Boosting Machine. *Annals of Statistics* 29 (2000), 1189–1232.

[13] Björn Gambäck and Utpal Kumar Sikdar. 2017. Using Convolutional Neural Networks to Classify Hate-Speech. In *Proceedings of the First Workshop on Abusive Language Online*. Association for Computational Linguistics, 85–90. https://doi.org/10.18653/v1/W17-3013

[14] Yoav Goldberg. 2016. A Primer on Neural Network Models for Natural Language Processing. *J. Artif. Intell. Res.* 57 (2016), 345–420.

[15] Ian J. Goodfellow, Yoshua Bengio, and Aaron C. Courville. 2016. *Deep Learning*. MIT Press. http://www.deeplearningbook.org/

[16] Edouard Grave, Tomas Mikolov, Armand Joulin, and Piotr Bojanowski. 2017. Bag of Tricks for Efficient Text Classification. In *EACL 2017*. 427–431.

[17] Nir Grinberg, Kenneth Joseph, Lisa Friedland, Briony Swire-Thompson, and David Lazer. 2019. Fake news on Twitter during the 2016 U.S. presidential election. *Science* 363, 6425 (2019), 374–378. https://doi.org/10.1126/science.aau2706 arXiv:http://science.sciencemag.org/content/363/6425/374.full.pdf

[18] Tommi Gröndahl, Luca Pajola, Mika Juuti, Mauro Conti, and N. Asokan. 2018. All You Need is "Love": Evading Hate Speech Detection. In *Proceedings of the 11th ACM Workshop on Artificial Intelligence and Security, CCS 2018, Toronto, ON, Canada, October 19, 2018*. 2–12.

[19] Homa Hosseinmardi, Sabrina Arredondo Mattson, Rahat Ibn Rafiq, Richard Han, Qin Lv, and Shivakant Mishra. 2015. Detection of Cyberbullying Incidents on the Instagram Social Network. *CoRR* abs/1503.03909 (2015). arXiv:1503.03909 http://arxiv.org/abs/1503.03909

[20] Jihye Lee. 2018. Twitter Apologizes for Mishandling Reported Threat From Mail-Bomb Suspect. http://time.com/5436809/twitter-apologizes-threat-mail-bomb-suspect/. [Online; accessed 26-January-2019].

[21] Tomas Mikolov, Ilya Sutskever, Kai Chen, Gregory S. Corrado, and Jeffrey Dean. 2013. Distributed Representations of Words and Phrases and their Compositionality. In *NIPS 2013*. 3111–3119.

[22] Etienne Papegnies, Vincent Labatut, Richard Dufour, and Georges Linares. 2017. Graph-based Features for Automatic Online Abuse Detection. In *International Conference on Statistical Language and Speech Processing*. Springer, 70–81.

[23] Ji Ho Park and Pascale Fung. 2017. One-step and Two-step Classification for Abusive Language Detection on Twitter. In *Proceedings of the First Workshop on Abusive Language Online*. Association for Computational Linguistics, 41–45. https://doi.org/10.18653/v1/W17-3006

[24] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global Vectors for Word Representation. In *EMNLP 2014*. 1532–1543.

[25] Kelly Reynolds, April Kontostathis, and Lynne Edwards. 2011. Using Machine Learning to Detect Cyberbullying. In *10th International Conference on Machine Learning and Applications and Workshops, ICMLA 2011, Honolulu, Hawaii, USA, December 18-21, 2011. Volume 2: Special Sessions and Workshop*. 241–244.

[26] Semiu Salawu, Yulan He, and Joanna Lumsden. 2017. Approaches to Automated Detection of Cyberbullying: A Survey. *IEEE Transactions on Affective Computing* (2017).

[27] Anna Schmidt and Michael Wiegand. 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*. 1–10.

[28] Vivek K Singh, Souvick Ghosh, and Christin Jose. 2017. Toward multimodal cyberbullying detection. In *Proceedings of the 2017 CHI Conference Extended Abstracts on Human Factors in Computing Systems*. ACM, 2090–2099.

[29] Zeerak Waseem. 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*. 138–142.

[30] Zeerak Waseem and Dirk Hovy. 2016. Hateful Symbols or Hateful People? Predictive Features for Hate Speech Detection on Twitter. In *Proceedings of the Student Research Workshop, SRW@HLT-NAACL 2016, The 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, San Diego California, USA, June 12-17, 2016*. 88–93. http://aclweb.org/anthology/N/N16/N16-2013.pdf

[31] 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*. International World Wide Web Conferences Steering Committee, 1391–1399.

[32] Ziqi Zhang, David Robinson, and Jonathan A. Tepper. 2018. Detecting Hate Speech on Twitter Using a Convolution-GRU Based Deep Neural Network. In *The Semantic Web - 15th International Conference, ESWC 2018, Heraklion, Crete, Greece, June 3-7, 2018, Proceedings*. 745–760.

A APPENDIX

A.1 Probability of overlapping tweets after oversampling

In this section we compute the probability that a tweet from the "hateful" class appears in both test and train set if one performs the oversampling before the data splitting. Thus, assume that dataset $\mathcal{T}$ is the union of two sets, say $\mathcal{T}_{\text{hate}}$ and $\mathcal{T}_{\text{none}}$, and assume that one creates the new dataset (actually, a multiset) $\mathcal{T}'$ by replicating the data in $\mathcal{T}_{\text{hate}}$ three times. That is $\mathcal{T}' = \mathcal{T}_{\text{hate}_1} \cup \mathcal{T}_{\text{hate}_2} \cup \mathcal{T}_{\text{hate}_3} \cup \mathcal{T}_{\text{none}}$ where $\mathcal{T}_{\text{hate}_i}$ is just a copy of $\mathcal{T}_{\text{hate}}$. Now assume that one uses a

Table 7: Partitions detail for experiments in Table 4

	Partition 1		Partition 2		Partition 3	
	Hate	None	Hate	None	Hate	None
Train	4,099	8,636	3,441	9,122	3,504	9,408
Test	740	1,474	1,398	988	1,335	702

probability p to sample data to create a test and train sets from $\mathcal{T}'$ (p for test and $(1-p)$ for train). Consider an arbitrary tweet $t^\star$ in $\mathcal{T}_{\text{hate}}$. Since $t^\star$ appears three times in $\mathcal{T}'$ the probability that one copy of $t^\star$ appears in the test set and the other two copies appear in the train set is

$$3 \times p \times (1-p) \times (1-p).$$

Similarly, the probability that two copies of $t^\star$ appear in the test set and one appears in the train set is

$$3 \times p \times p \times (1-p).$$

Thus, the total probability that $t^\star$ appears simultaneously in both sets is the sum of both expressions above. In particular, if $p = 0.15$ then we obtain a probability of

$$3 \times 0.15 \times 0.85 \times 0.85 \;+\; 3 \times 0.15 \times 0.15 \times 0.85$$

which is approximately 0.38. That is, if we use a test-train split in the proportion 15%-85% after oversampling three times every tweet in the "hateful" class, then every "hateful" tweet has a 38% probability of belonging simultaneously to the test and the train sets.

A.2 Details on the partitions for Table 4

Table 7 shows some details on the partitions of not overlapping users between train and test set for the Waseem and Hovy's dataset. The description on how we generate every partition is as follows:

- Partition 1: The users in the training set are the most prolific users per class: top 1 for "racist" class, top 2 for "sexist" class, and top 1 for the "none" class. We completed the training set adding other prolific users that only belong to the "none" class. The rest of the users were included in the testing set.
- Partition 2: In this partition, the second most prolific "sexist" user was included in the testing set. Then, the users composing the training set are: top 1 for "racist" class, top 1 for "sexist" class and top 1 for the "none" class. We completed the training set adding users belonging only to the "none" and "sexist" class.
- Partition 3: The training set is composed by all the "sexist" users except for the most prolific one, the top 1 most prolific "racist" user and other prolific users that only belong to the "none" class.