Reproducibility Report
Contextualizing Hate Speech Classifiers with Post-hoc Explanation

Anonymous Author(s)
Affiliation
Address
email

Reproducibility Summary

The presented report evaluates Contextualizing Hate Speech Classifiers with Post-hoc Explanation Kennedy et al. (2020) paper within the scope of ML Reproducibility Challenge 2020. Our work focuses on both aspects constituting the paper: the method itself and the validity of the stated results. In the following sections, we have described the paper, related works, algorithmic frameworks, our experiments and evaluations.

Scope of Reproducibility

For the GHC (a dataset), the most marked difference between BERT+WR and BERT+SOC is increased recall. While, for Stormfront (a dataset), we see the same improvements for in-domain data and NYT. For verifying the claims we also have tried to run the same experiment on a new data-set.

Methodology

We have tried to re-implement the author's code and verify the claims made in their original paper. We have experimented on NVIDIA Tesla GPU which was less efficient than the original author's resource (NVIDIA GeForceRTX 2080 Ti).

Results

We have able to reproduced claims as mentioned in the following section 2 (Scope of Reproducibility) marked as point 2 and 3. But we are not in the same page with the authors for a few reported experiments mentioned as point 3 and 4 in the same section.

What was easy

The original authors provide code for most of the experiments presented in the paper. The code was easy to run and allowed us to verify the correctness of our re-implementation. The explanations in the code made the work pretty easy for us.

What was difficult

Training the models was very time taking as we had to wait for hours to get that working and the resources the original authors used is not readily available everywhere.

Communication with original authors

We were in contact with the second author via E-mail, as he was responsive and shared details which were not explicitly mentioned in the paper.

Submitted to ML Reproducibility Challenge 2020. Do not distribute.

1 Introduction

Detection of hate speech is part of the continuous work to limit the harm done by oppressive and offensive language Waldron (2012); Gelber and McNamara (2016). Performance has increased with access to more data, and extra sophisticated algorithms Mondal et al. (2017); Silva et al. (2016), while the relative sparsity of hate speech needs sampling using keywords Olteanu et al. (2018) or a selection from environments with very high rates of hate speech de Gibert et al. (2018). Thus, modern text classifiers struggles to learn a model of hateful speech that generalizes to real-world applications Wiegand et al. (2019). A particular problem found in neural hate speech classifiers is their over-sensitivity to group identifiers like "Jews", "black", and "gay", which are only hate speech when connected with the right context Dixon et al. (2018). Neural text classifiers produce a state-of-the-art performance in hate speech detection but are uninterpretable also can break when supplied with unexpected inputs Niven and Kao (2019). It isn't easy to contextualize a model's method of identifying words. Author's approach to this issue is to use the Sampling and Occlusion (SOC) explanation algorithm, which estimates model agnostic, context-independent post-hoc feature importance Jin et al. (2020). They used this approach to the Gab Hate Corpus Kennedy et al. (2020), a new corpus labeled for "hate-based rhetoric", and an annotated corpus from the Stormfront the largest online community of white nationalists, characterised by pseudo-rational discussions of race de Gibert et al. (2018). On the basis of information generated via SOC, which showed models were biased towards group identifiers, they then propose a new regularization-based approach to improve model sensitivity to the context surrounding group identifiers. They regularize the importance of group identifiers at training, coercing models to analyze the context surrounding them. They noticed that regularization decreases the attention given to group identifiers and increases the importance of the more generalizable features of hate speech, like dehumanizing and abusive language. In experiments on an out-of-domain test set of news articles comprising group identifiers, which are heuristically expected to be non-hate speech, they discover that regularization dramatically reduces the false positive rate. At the same time, in-domain, out-of-sample classification performance is either maintained or improved.

2 Scope of reproducibility

The paper here points out that most of the Hate Speech classifiers available now are majorly tilted or over-sensitive to some of the identifiers or words like (gay, black, and Muslim) but they don't take into account the fact that the mere presence of the word would not make it oppressive but the context in which it is used gives us the correct classification. If the context is not taken into account then many samples would result in false positives. Thereby, reducing the accuracy. The work here is formulated to detect hate speech as disambiguating the use of offensive words from abusive versus non-abusive contexts. We plan to use the code that is available from the authors themselves and then as per the paper we will be reviewing and testing the claims made. Some of the major claims of the paper are:

1. For the GHC (a dataset), the most marked difference between BERT+WR and BERT+SOC is increased recall
2. For Stormfront (a dataset), we see the same improvements for in-domain data and NYT.
3. Paper claims performance for their proposed method as (Precision = 56.11, Recall = 54.23, F1 = 54.71 and NYT Acc = 93.89) on average
4. The efficiency claimed in the paper is as follows (BERT = 5:1 mins, BERT+OC = 13:36 mins, BERT+SOC = 19:3 mins)

We have tried to verify the above claims made in the paper using the data-sets presented by the original authors and as well as on a new data-set. To train the model the authors have used GeForce RTX 2080 Ti GPU, which we tried to implement using our institutional resources.

3 Methodology

The authors in their previous paper Jin et al. (2020) have explained methods which are used in the current paper. We here first explain the parts of the previous paper and then show how it is used in the current paper. The methods and approach is described below:

3.1 Model descriptions

3.1.1 Context-Independent Importance (CII)

Given a phrase p := $x_{i:j}$ appearing in a specific input $x_{1:T}$, first the setting is relaxed and then they define the importance of a phrase independent of contexts of length N adjacent to it. For an intuitive example, to evaluate the CII up to one

word of very in the sentence The film is very interesting in a sentiment analysis model, then sample some possible adjacent words, and average the prediction difference after some practice of masking the word very (as shown in Figure 1 below). The N-context independent importance is formally written in Equation 1.

$$\phi(p, \hat{x}) = E_{x_\delta}[s(x_{-\delta}; \hat{x}_\delta) - s(x_{-\delta} \backslash p; \hat{x}_\delta)] \tag{1}$$

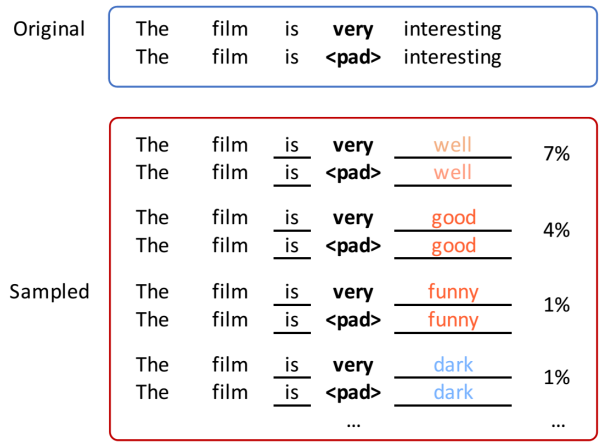

Figure 1: Word Masking and Value Prediction

where $x_{-\delta}$ denotes the resulting sequence after masking out a context of length N surrounding the phrase p from the input x. Here, $\hat{x}_\delta$ is a sequence of length N sampled from a distribution $p(\hat{x}_\delta | x_{-\delta})$, which is conditioned on the phrase p as well as other words in the sentence x. Accordingly, they use $s(x_{-\delta}; \hat{x}_\delta)$ to denote the model prediction score after replacing the masked-out context $x_{-\delta}$ with a sampled context $\hat{x}_\delta$. $x \backslash p$ is used to denote the operation of masking out the phrase p from the input sentence x. Following the notion of N-CII, they define CII of a phrase p by increasing the size of the context N to sufficiently large (e.g., length of the sentence). The CII can be equivalently written as given in Equation 2.

$$\phi^g(p) = E_x[s(x) - s(x \backslash p) | p \subseteq x] \tag{2}$$

muslim jew jews white islam blacks muslims women whites gay black democat islamic allah jewish lesbian transgender race brown woman mexican religion homosexual homosexuality africans

Figure 2: 25 group identifiers selected from top weighted words in the TF-IDF BOW linear classifier on the GHC

3.1.2 Model Interpretation

To determine the problem more quantitatively, they examine hate speech models' bias towards group identifiers and how this leads to false-positive errors during prediction. Then they determine a model's sensitivity towards group identifiers by examining the models themselves. Linear classifiers can be examined in terms of their most highly-weighted features. Then by applying a post-hoc explanation algorithm for this task of extracting similar information from the fine-tuned methods discussed above. From the top features in a bag-of-words logistic regression of hate speech on GHC_{train}, they collected a set of twenty-five identity words which are used in later analyses.

Explanation-based measures: BERT models are able to model complicated word and phrase compositions: for example, some words are only offensive when they are composed with specific ethnic groups. To capture this, Sampling and Occlusion (SOC) algorithm is used which is capable of generating hierarchical explanations for a prediction. SOC starts by assigning importance score for phrases in a way that eliminates compositional effect between the phrase and its context x_δ around it. Given a phrase p appearing in a sentence x, SOC assigns an importance score $\phi(p)$ to show how the phrase p contribute so that the sentence is classified as a hate speech. The algorithm computes the difference of the unnormalized prediction score $s(x)$ between "hate" and "non-hate" in the 2-way classifier. Then the algorithm

evaluates average change of $s(x)$ when the phrase is masked with padding tokens (noted as $x \backslash p$) for different inputs, in which the N-word contexts around the phrase p are sampled from a pretrained language model, while other words remain the same as the given x. Formally, the importance score $\phi(p)$ is measured as given in Equation 3.

$$\phi(p) = E_{x_\delta}[s(x) - s(x \backslash p)] \tag{3}$$

In the meantime, SOC algorithm perform agglomerative clustering over explanations to generate a hierarchical layout. Using SOC explanations output on GHC_{test}, they compute average word importance and present the top 20 in Figure 4.

jew jews mexican blacks jewish brown black muslim homosexual islam

Figure 3: 10 group identifiers selected for the Stormfront dataset

Bias in Prediction: Hate speech models can be over-attentive to group identifiers. To provide an external measure of models' over-sensitivity to group identifiers, they construct an adversarial test set of New York Times (NYT) articles that are filtered to contain a balanced, random sample of the twenty-five (GHC Dataset) and ten (Stromfront dataset) group identifiers as shown in Figure 2 and Figure 3 respectively.

BERT	**Δ Rank**	**Reg.**	**Δ Rank**
ni**er	+0	ni**er	+0
ni**ers	-7	fag	+35
kike	-90	traitor	+38
mosques	-260	faggot	+5
ni**a	-269	bastard	+814
jews	-773	blamed	+294
kikes	-190	alive	+1013
nihon	-515	prostitute	+56
faggot	+5	ni**ers	-7
nip	-314	undermine	+442
islam	-882	punished	+491
homosexuality	-1368	infection	+2556
nuke	-129	accusing	+2408
niro	-734	jaggot	+8
muhammad	-635	poisoned	+357
faggots	-128	shitskin	+62
nitrous	-597	ought	+229
mexican	-51	rotting	+358
negro	-346	stayed	+5606
muslim	-1855	destroys	+1448

Figure 4: Top 20 words by mean SOC weight before (BERT) and after (Reg.) regularization for GHC

It is key for models to not ignore identifiers, but to match them with the right context. Figure 5 shows the effect of ignoring identifiers: random subsets of words ranging in size from 0 to 25 are removed, with each subset sample size repeated 5 times. Decreased rates of false positives on the NYT set are accompanied by poor performance in hate speech detection

Explanation Regularization: Given that SOC explanations are fully differentiable, during training, they regularize SOC explanations on the group identifiers to be close to 0 in addition to the classification objective $\mathcal{L}'$. The combined learning objective is written in Equation 4.

$$\mathcal{L} = \mathcal{L}' + \alpha \sum_{w \in x \cap S} [\phi(w)]^2 \tag{4}$$

where S notes for the set of group names and x notes for the input word sequence. α is a hyper-parameter for the strength of the regularization. In addition to SOC, they also experiment with regularizing input occlusion (OC) explanations, defined as the prediction change when a word or phrase is masked out, bypassing the sampling step in SOC.

Visualizing Effects of Regularization: It can be further seen the effect of regularization by considering Figure 5, where hierarchically clustered explanations from SOC are visualized before and after regularization, correcting a false positive.

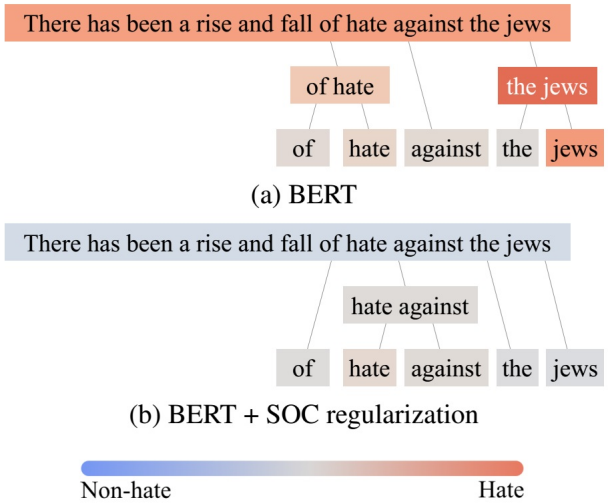

Figure 5: Hierarchical explanations on a test instance from GHCtest before and after explanation regularization, where false positive predictions are corrected

3.2 Datasets

We chose two public corpora for our experiments which feature the logical parts of hate speech, versus only the use of slurs and explicitly hostile language (see Davidson et al., 2017). The "Gab Hate Corpus" (GHC; Kennedy et al., 2020) is a huge, arbitrary example (N = 27,655) from the Pushshift.io information dump of the Gab network, which we have annotated on as per a typology of "hate based manner of speaking", a build motivated by hate speech criminal codes outside the U.S. also, sociology research on bias and dehumanization. Gab is a social network with a high pace of hate discourse (Zannettou et al., 2018; Lima et al., 2018) and populated by the "Extreme right" (Anthony, 2016; Benson, 2016). Likewise with deference to area and definitions, de Gibert et al. (2018) sampled and annotated posts from the "Stormfront" web space (Meddaugh and Kay, 2009) and annotated at the sentence level as per a comparable annotation guide as utilized in the GHC.

Table 1: GHC Dataset

GHC	Total	Hate	Non Hate
Train	24353	2027	22326
Test	1586	372	1214

Table 2: Stormfront Dataset

Stormfront	Total	Hate	Non Hate
Train	7896	1059	6837
Test	1998	246	1752
Validation	979	122	857

Table 3: New (Twitter) Dataset

Twitter Dataset	Total	Hate	Non Hate
Train	6555	780	5775
Test	1634	196	1438
Validation	1156	140	1016

Train and test parts were arbitrarily produced for Stormfront sentences (80/20) with "hate" taken as positive binary label, and a test set was compiled from the GHC by drawing random stratified data regarding the "target population" tag (potential qualities including race/identity target, sexual religious and so forth) A solitary "hate" mark was made by taking the association of two fundamental labels, “human degradation” and “calls for violence”. Training data for the (GHC train) included 24,353 posts with 2,027 marked as hate, and test data for the (GHC test) included 1,586 posts with 372 marked as hate. In the Stormfront dataset out of 7,896 posts, 1,059 marked as hate in training sentences, out of 979 posts, 122 marked as hate in the validation set, and out of 1,998 posts, 246 marked as hate in the test dataset. We have trained the model on our new Twitter dataset taken from Kaggle [1]. Training data for the (new train) included 6555 posts with 780 marked as hate, test data for the (new test) included 1634 posts with 196 marked as hate, and validation data for the (new val) included 1156 posts with 140 marked as hate. Table 1 shows the number of hate and non hate labels in the GHC Dataset. Table 2 shows the number of hate and non hate labels in the Stormfront dataset. Table 3 shows the number of hate and non hate labels in the Twitter dataset. The Figure 6 shows the comparison of old vs new dataset.

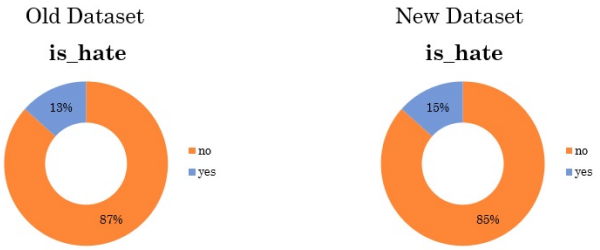

Figure 6: Old(Stormfront) vs New(Twitter) Dataset Comparison

3.3 Computational requirements

Provide information on computational requirements for each of your experiments. For example, the number of CPU/GPU hours and memory requirements. You’ll need to think about this ahead of time, and write your code in a way that captures this information so you can later add it to this section. The authors have reported that they have used GeForce RTX 2080 Ti GPU to train each model and the the training times of BERT+SOC and BERT+OC were only 4 times and 2 times of the original BERT. We have experimented on NVIDIA Tesla GPU. The detail comparisons of GPU and time are shown in Table 4 and 5. The authors framework were far superior to ours which may be the reason that their training time and usage are more efficient than ours.

Table 4: GPU Comparisons

GPU Features	Paper Report	Our Framework
GPU Name	TU102	GK110B
GPU Details	NVIDIA GeForce RTX 2080 Ti	NVIDIA Tesla K40m
Memory Size	11 GB	12 GB
Memory Clock	14 Gbps	6 Gbps
Memory Type	GDDR6	GDDR5

4 Results

We have investigated different methods BERT, Word identifiers removal before BERT training (BERT+WR), BERT with regularizing occlusion (BERT+OC) and BERT with regularizing sampling and occlusion (BERT+SOC) with similar parameter and hyper-parameter values as described by the authors. We also used the NYT test set as blind dataset for quantitative measure of how well a model has learnt the contexts in which group identifiers are used for hate speech. We experimented on the GHC, Stormfront and external labelled Twitter dataset for evaluating hate speech classification in-domain and accuracy on the NYT test set. Similar to the paper, we have used the full list of 25 terms (for GHC); 10 terms for Stormfront. Congruently, for Stormfront we filtered the NYT corpus to contain these 10 terms (N = 5,000).

[1] https://www.kaggle.com/vkrahul/twitter-hate-speech

Table 5: Time Comparisons

Methods	Approach	Training Time (per epoch)	GPU memory use
BERT	Paper	5 m 1 s	9095M
	Ours	15 m 13 s	7253M
BERT + OC	Paper	12 m 36 s	9411M
	Ours	21 m 5 s	7041M
BERT + SOC	Paper	19 m 3 s	9725M
	Ours	24 m 33 s	7352M

Table 6: Precision, Recall, F1-score (%) on Stormfront (Stf.) test set and accuracy (%) on NYT evaluation set.

Training Set		Stromfront			
Method / Metrics	Approach	Precision	Recall	F1-Score	NYT Accuracy
BERT	Paper	57.76 ± 3.9	54.43 ± 8.1	55.44 ± 2.9	92.29 ± 4.1
	Ours	54.23	59.81	56.88	91.14
BERT+WR	Paper	53.16 ± 4.3	57.16 ± 5.7	54.60 ± 1.7	92.47 ± 3.4
	Ours	56.74	53.62	55.14	94.59
BERT + OC (α = 0.1)	Paper	57.47 ± 3.7	51.10 ± 4.4	53.82 ± 1.3	95.39 ± 2.3
	Ours	57.6	59.01	58.29	89.47
BERT + SOC (α = 1.0)	Paper	56.05 ± 3.7	54.35 ± 3.4	54.97 ± 1.1	95.40 ± 2.0
	Ours	65.26	50.82	57.15	90.05

Performances (as reported in this paper and what we obtain during reproducibility experiment) are shown in Table 6 and Table 7 for Stromfront and GHC dataset respectively. We report mean and standard deviation of the performance across 10 runs for BERT, BERT + WR (word removal), BERT + OC, and BERT + SOC (as reported in paper) and we have tested the reproduced results. Though our reproduced results are comparable as per reported in the paper for most of the methods in Sromfront datasets but the we obtain lower precision, recall and F1-score for GHC dataset (BERT+SOC with $\alpha = 0.1$). Testing on blind dataset NYT is comparable for most of the cases. Only in a few cases our reproduced results differ from paper reported range values for Stromfront dataset like higher precision (+ 9%) and lower accuracy (- 5%) in BERT+SOC ($\alpha = 1.0$).

Table 7: Precision, Recall, F1-Score (%) on GHC test set and accuracy (%) on NYT evaluation set for BERT + SOC ($\alpha = 0.1$)

Training	Method	BERT + SOC ($\alpha = 0.1$)	
Set	Metrics	Paper	Our Repro
GHC	Precision	60.56 ± 1.8	41.66
	Recall	69.72 ± 3.6	58.14
	F1-Score	64.14 ± 3.2	48.54
	NYT Accuracy	89.43 ± 4.3	89.97

Table 8: Precision, Recall, F1-Score (%) on New Twitter test set and accuracy (%) on NYT evaluation set for BERT + SOC ($\alpha = 1.0$)

Training	Method	BERT + SOC ($\alpha = 1.0$)
Set	Metrics	Ours
Twitter Test Set	Precision	80.61 ± 3.9
	Recall	56.42 ± 3.4
	F1-Score	66.38 ± 1.1

Table 8 shows precision, recall, f1-score and NYT-accuracy obtained by BERT+SOC ($\alpha = 1.0$) method on new Twitter hate corpus. Comparisons of different metrics on new Twitter dataset and Stromfront dataset are shown in Figure 7 which shows significant increase of precision (20%) on new dataset by BERT+SOC (α=1.0) algorithm than Stromfront.

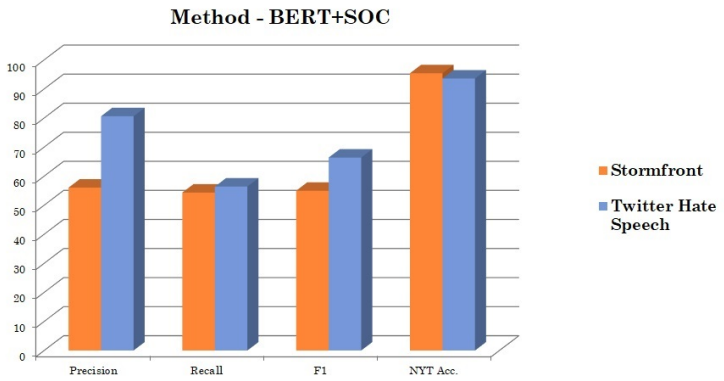

Figure 7: Comparison of Metrics for BERT+SOC (α=1.0) between Stromfront and Twitter Dataset

5 Discussion

We have able to reproduced few claims as reported by the authors in the paper - (i) For Stormfront (a dataset), we see the same improvements for in-domain data and NYT and (ii) Paper claims performance for their proposed method as (Precision = 56.11, Recall = 54.23, F1 = 54.71 and NYT Acc = 93.89) on average. But we are not in the same page with the authors for a few reported experiments - for the GHC, the most marked difference between BERT+WR and BERT+SOC is increased recall as we have obtained lower precision, recall and f1-accuracy. This may be due the experimental framework differences. Due to a bar on time, we could not run the BERT+SOC several times to make the comparison more detailed. In the future, we would also try to verify their claims using similar GPU configurations and incorporate more new datasets.

5.1 What was easy

The authors' code which was publicly available, covered almost all the experiments in their paper it also helped us to validate the correctness of our replicated codebase. Additionally, the original paper is quite complete, straightforward to follow, and the ReadMe file in their project helped a lot.

5.2 What was difficult

For replicating the experiments one will need the GPU of similar comparison as used by the original authors or it will be difficult to get results on time as was in our case.

5.3 Communication with original authors

While working on the challenge, we stood in E-mail contact with the second author and want to thank the author for his responsive communication, which helped clarify a great deal of implementation and evaluation specifics. For example, which particular BERT model from library was used to train the model. We also got the data-sets that they were using for carrying out the experiments. The communication helped a lot in understanding the paper.

References

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

Jeremy Waldron. 2012. *The harm in hate speech*. Harvard University Press.

Katharine Gelber and Luke McNamara. 2016. Evidencing the harms of hate speech. *Social Identities*, 22(3):324–341.

Mainack Mondal, Leandro Araújo Silva, and Fabrício Benevenuto. 2017. A measurement study of hate speech in social media. In *Proceedings of the 28th ACM Conference on Hypertext and Social Media*, pages 85–94. ACM.

Leandro Silva, Mainack Mondal, Denzil Correa, Fabrício Benevenuto, and Ingmar Weber. 2016. Analyzing the targets of hate in online social media. In *Tenth International AAAI Conference on Web and Social Media*.

Alexandra Olteanu, Carlos Castillo, Jeremy Boy, and Kush R Varshney. 2018. The effect of extremist violence on hateful speech online. In *Twelfth International AAAI Conference on Web and Social Media*.

Ona de Gibert, Naiara Perez, Aitor García Pablos, and Montse Cuadros. 2018. Hate speech dataset from a white supremacy forum. In *Proceedings of the 2nd Workshop on Abusive Language Online (ALW2)*, pages 11–20.

Michael Wiegand, Josef Ruppenhofer, and Thomas Kleinbauer. 2019. Detection of abusive language: the problem of biased datasets. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 602–608.

Lucas Dixon, John Li, Jeffrey Sorensen, Nithum Thain, and Lucy Vasserman. 2018. Measuring and mitigating unintended bias in text classification. In *Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society*, pages 67–73. ACM.

Timothy Niven and Hung-Yu Kao. 2019. Probing neural network comprehension of natural language arguments. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4658–4664.

Xisen Jin, Zhongyu Wei, Junyi Du, Xiangyang Xue, and Xiang Ren. 2020. Towards hierarchical importance attribution: Explaining compositional semantics for neural sequence models. In *International Conference on Learning Representations*.

Brendan Kennedy, Mohammad Atari, Aida M Davani, Leigh Yeh, Ali Omrani, Yehsong Kim, Kris Coombs Jr., Shreya Havaldar, Gwenyth Portillo-Wightman, Elaine Gonzalez, Joe Hoover, Aida Azatian, Gabriel Cardenas, Alyzeh Hussain, Austin Lara, Adam Omary, Christina Park, Xin Wang, Clarisa Wijaya, Yong Zhang, Beth Meyerowitz, and Morteza Dehghani. 2020. The gab hate corpus: A collection of 27k posts annotated for hate speech.