

HateXplain: A Benchmark Dataset for Explainable Hate Speech Detection*

Binny Mathew^{1†}, Punyajoy Saha^{1†}, Seid Muhie Yimam²
Chris Biemann², Pawan Goyal¹, Animesh Mukherjee¹

¹ Indian Institute of Technology, Kharagpur, India

² Universität Hamburg, Germany

binnyathew@iitkgp.ac.in, punyajoy@iitkgp.ac.in, yimam@informatik.uni-hamburg.de
biemann@informatik.uni-hamburg.de, pawang@cse.iitkgp.ac.in, animeshm@cse.iitkgp.ac.in

Abstract

Hate speech is a challenging issue plaguing the online social media. While better models for hate speech detection are continuously being developed, there is little research on the *bias* and *interpretability* aspects of hate speech. In this paper, we introduce HateXplain, the first benchmark hate speech dataset covering multiple aspects of the issue. Each post in our dataset is annotated from three different perspectives: the basic, commonly used 3-class *classification* (i.e., hate, offensive or normal), the *target community* (i.e., the community that has been the victim of hate speech/offensive speech in the post), and the *rationales*, i.e., the portions of the post on which their labelling decision (as hate, offensive or normal) is based. We utilize existing state-of-the-art models and observe that even models that perform very well in classification do not score high on explainability metrics like model *plausibility* and *faithfulness*. We also observe that models, which utilize the human rationales for training, perform better in reducing unintended bias towards target communities. We have made our code and dataset public¹ for other researchers.

Disclaimer: The article contains material that many will find offensive or hateful; however this cannot be avoided owing to the nature of the work.

Introduction

The increase in online hate speech is a major cultural threat, as it already resulted in crime against minorities, see e.g. (Williams et al. 2020). To tackle this issue, there has been a rising interest in hate speech detection to expose and regulate this phenomenon. Several hate speech datasets (Ousidhoum et al. 2019; Qian et al. 2019b; de Gibert et al. 2018; Sanguinetti et al. 2018), models (Zhang, Robinson, and Tepper 2018; Mishra et al. 2018; Qian et al. 2018b,a), and shared tasks (Basile et al. 2019; Bosco et al. 2018) have been made available in the recent years by the community, towards the development of automatic hate speech detection.

While many models have claimed to achieve state-of-the-art performance on some datasets, they fail to generalize (Arango, Pérez, and Poblete 2019; Gröndahl et al. 2018). The models may classify comments that refer to certain commonly-attacked identities (e.g., gay, black, muslim)

as toxic without the comment having any intention of being toxic (Dixon et al. 2018; Borkan et al. 2019). A large prior on certain trigger vocabulary leads to biased predictions that may discriminate against particular groups who are already the target of such abuse (Sap et al. 2019; Davidson, Bhattacharya, and Weber 2019). Another issue with the current methods is the lack of explanation about the decisions made. With hate speech detection models becoming increasingly complex, it is getting difficult to explain their decisions (Goodfellow, Bengio, and Courville 2016). Laws such as General Data Protection Regulation (GDPR (Council 2016)) in Europe have recently established a “right to explanation”. This calls for a shift in perspective from performance based models to interpretable models. In our work, we approach model explainability by learning the target classification and the reasons for the human decision jointly, and also to their mutual improvement.

We therefore have compiled a dataset that covers multiple aspects of hate speech. We collect posts from Twitter² and Gab³, and ask Amazon Mechanical Turk (MTurk) workers to annotate these posts to cover three facets. In addition to classifying each post into hate, offensive, or normal speech, annotators are asked to select the target communities mentioned in the post. Subsequently, the annotators are asked to highlight parts of the text that could justify their classification decision⁴. The notion of justification, here modeled as ‘human attention’, is very broad with many possible realizations (Lipton 2018; Doshi-Velez 2017). In this paper, we specifically focus on using *rationales*, i.e., snippets of text from a source text that support a particular categorization. Such rationales have been used in common-sense explanations (Rajani et al. 2019), e-SNLI (Camburu et al. 2018) and several other tasks (DeYoung et al. 2020). If these rationales are good reasons for decisions, then models guided towards these in training could be made more human-decision-taking-like.

Consider the examples in Table 1. The first row shows the tokens (‘rationales’) that were selected by human annotators which they believe are important for the classifi-

* Accepted at AAAI 2021.

[†] Equal Contribution

¹ <https://github.com/punyajoy/HateXplain>

² <https://twitter.com/>

³ <https://gab.com/>

⁴ In case the post is classified as normal, the annotators does not need to highlight any span.

Model	Text	Label
Human Annotator	The jews are again using holohoax as an excuse to spread their agenda . Hilter should have eradicated them	HS
CNN-GRU	The jews are again using holohoax as an excuse to spread their agenda . Hilter should have eradicated them	HS
BiRNN	The jews are again using holohoax as an excuse to spread their agenda . Hilter should have eradicated them	HS
BiRNN-Attn	The jews are again using holohoax as an excuse to spread their agenda . Hilter should have eradicated them	HS
BiRNN-HateXplain	The jews are again using holohoax as an excuse to spread their agenda . Hilter should have eradicated them	HS
BERT	The jews are again using holohoax as an excuse to spread their agenda . Hilter should have eradicated them	OF
BERT-HateXplain	The jews are again using holohoax as an excuse to spread their agenda . Hilter should have eradicated them	OF

Table 1: Example of the rationales predicted by different models compared to human annotators. The **green highlight** marks tokens that the human annotator and model found important for the prediction. The **orange highlight** marks tokens which the model found important, but the human annotators did not.

cation. The next six rows show the important tokens (using LIME (Ribeiro, Singh, and Guestrin 2016)), which helped various models in the classification. We observe that even when the model is making the correct prediction (hate speech – HS in this case), the reason (‘rationales’) for this varies across models. In case of BERT, we observe that it attends to several of the tokens that human annotators deemed important, but assigns the wrong label (offensive speech - OF).

In summary, we introduce **HateXplain**, the first benchmark dataset for hate speech with word and phrase level span annotations that capture human rationales for the labeling. Using MTurk, we collect a large dataset of around 20K posts and annotate them to cover three aspects of each post. We use several models on this dataset and observe that while they show a good model performance, they do not fare well in terms of model interpretability/explainability. We also observe that providing these rationales as input during training helps in improving a model’s performance and reducing the unintended bias. We believe that this dataset would serve as a fundamental source for the future hate speech research.

Related work

Hate speech

The public expression of hate speech affects the devaluation of minority members (Greenberg and Pyszczynski 1985) and such frequent and repetitive exposure to hate speech could increase an individual’s outgroup prejudice (Soral, Bilewicz, and Winiewski 2018). Real world violent events could also lead to increased hate speech in online space (Olteanu et al. 2018). To tackle this, various methods have been proposed for hate speech detection (Burnap and Williams 2016; Ribeiro et al. 2018; Zhang, Robinson, and Tepper 2018; Qian et al. 2018a). The recent interest in hate speech research has led to the release of datasets in multiple languages (Ousidhoum et al. 2019; Sanguinetti et al. 2018) along with different computational approaches to combat online hate (Qian et al. 2019a; Mathew et al. 2019b; Aluru et al. 2020).

A recurrent issue with the majority of previous research is that many of them tend to conflate hate speech and abusive/offensive⁵ language (Davidson et al. 2017). Some of the

works have combined offensive and hate language under a single concept, while very few works, such as (Davidson et al. 2017; Founta et al. 2018) and Van Huynh et al. (2019) have attempted to separate offensive from hate speech. We argue that this, although subjective, is an important aspect as there are lots of messages that are offensive but do not qualify as hate speech. For example, consider the word ‘*nigga*’. The word is used everyday in online language by the African American community (Vigna et al. 2017). Similarly, words like *hoe* and *bitch* are used commonly in rap lyrics. Such language is prevalent on social media (Wang et al. 2014) and any hate speech detection system should include these for the system to be usable. To this end, we have assumed that a given text can belong to one of the three classes: hate, offensive, normal. We have adopted the classes based on the work of Davidson et al. (2017). Table 2 provides a comparison between some hate speech datasets.

Explainability/Interpretability

Zaidan, Eisner, and Piatko (2007) introduced the concept of using *rationales*, in which human annotators would highlight a span of text that could support their labeling decision. The authors utilized these enriched rationale annotation on a smaller set of training data, which helped to improve sentiment classification. Yessenalina, Choi, and Cardie (2010) built on this work and developed methods that automatically generate rationales. Lei, Barzilay, and Jaakkola (2016) also proposed an encoder-generator framework, which provides quality rationales without any annotations.

In our paper, we utilize the concept of *rationales* and provide the first benchmark hate speech dataset with human level explanations. We have made our model and dataset public¹ for other researchers.

Dataset collection and annotation strategies

In this section, we provide the annotation strategies we have followed, the dataset selection approaches used, and the statistics of the collected dataset.

Dataset sampling

We collect our dataset from sources where previous studies on hate speech have been conducted: **Twitter** (Davidson

¹in our paper as they are arguably very similar (Founta et al. 2018).

⁵We have used the terms offensive and abusive interchangeably

Dataset	Labels	Total Size	Language	Target Labels?	Rationales?
Waseem and Hovy (2016)	racist, sexist, normal	16,914	English	✗	✗
Davidson et al. (2017)	Hate Speech, Offensive, Normal	24,802	English	✗	✗
Founta et al. (2018)	Abusive, Hateful, Normal, Spam	80,000	English	✗	✗
Ousidhoum et al. (2019)	Labels for five different aspects	13,000	English, French, Arabic	✓	✗
HateXplain (Ours)	Hate Speech, Offensive, Normal	20,148	English	✓	✓

Table 2: Comparison of different hate speech datasets.

et al. 2017; Fortuna and Nunes 2018) and **Gab** (Lima et al. 2018; Mathew et al. 2020; Zannettou et al. 2018). Following the existing literature, we build a corpus of posts (tweets and gab posts) using lexicons. We combined the lexicon set provided by Davidson et al. (2017), Ousidhoum et al. (2019), and Mathew et al. (2019a) to generate a single lexicon. For Twitter, we filter the tweets from the 1% randomly collected tweets in the time period Jan-2019 to Jun-2020. In case of Gab, we use the dataset provided by Mathew et al. (2019a). We do not consider reposts and remove duplicates. We also ensure that the posts do not contain links, pictures, or videos as they indicate additional information that might not be available to the annotators. However, we do not exclude the emojis from the text as they might carry important information for the hate and offensive speech labeling task. The posts were anonymized by replacing the usernames with <user>token.

Annotation procedure

We use Amazon Mechanical Turk (MTurk) workers for our annotation task. Each post in our dataset contains three types of annotations. First, whether the text is a hate speech, offensive speech, or normal. Second, the target communities in the text. Third, if the text is considered as hate speech, or offensive by majority of the annotators, we further ask the annotators to annotate parts of the text, which are words or phrases that could be a potential reason for the given annotation. These additional span annotations allow us to further explore how hate or offensive speech manifests itself.

Target group annotation The primary goal of the annotation task is to determine whether a given text is hateful, offensive, or neither of the two, i.e. normal. As noted above, we also get span annotations as reasons for the label assigned to a post (hateful or offensive). To further enrich the dataset, we ask the workers to decide the groups that the hate/offensive speech is targeting. Table 3 lists the target groups we have identified.

Target groups	Categories
Race	African, Arabs, Asians, Caucasian, Hispanic
Religion	Buddhism, Christian, Hindu, Islam, Jewish
Gender	Men, Women
Sexual Orientation	Heterosexual, LGBTQ
Miscellaneous	Indigenous, Refugee/Immigrant, None, Others

Table 3: Target groups considered for the annotation.

Annotation instructions and design of the interface Before starting the annotation task, workers are explicitly

	Twitter	Gab	Total
Hateful	708	5,227	5,935
Offensive	2,328	3,152	5,480
Normal	5,770	2,044	7,814
Undecided	249	670	919
Total	9,055	11,093	20,148

Table 4: Dataset details. “Undecided” refers to the cases where all the three annotators chose a different class.

warned that the annotation task displays some hateful or offensive content. We prepare instructions for workers that clearly explain the goal of the annotation task, how to annotate spans and also include a definition for each category. We provided multiple examples with classification, target community and span annotations to help the annotators understand the task. To further ensure high quality dataset, we use built-in MTurk qualification requirements, namely the *HIT Approval Rate (95%) for all Requesters’ HITs* and the *Number of HITs Approved (5,000)* requirements.

Dataset creation steps

For the dataset creation, we first conducted a pilot annotation study followed by the main annotation task.

Pilot annotation: In the pilot task, each annotator was provided with 20 posts and they were required to do the hate/offensive speech classification as well as identify the target community (if any). In order to have a clear understanding of the task, they were provided with multiple examples along with explanations for the labelling process. The main purpose of the pilot task was to shortlist those annotators who were able to do the classification accurately. We also collected feedback from annotators to improve the main annotation task. A total of 621 annotators took part in the pilot task. Out of these, 253 were selected for the main task.

Main annotation: After the pilot annotation, once we had ascertained the quality of the annotators, we started with the main annotation task. In each round, we would select a batch of around 200 posts. Each post was annotated by three annotators, then majority voting was applied to decide the final label. The final dataset is composed of 9,055 posts from Twitter and 11,093 posts from Gab. Table 4 provides further details about the dataset collected. Table 5 shows samples of our dataset. The Krippendorff’s α for the inter-annotator agreement is 0.46 which is much higher than other hate speech datasets (Vigna et al. 2017; Ousidhoum et al. 2019).

Class labels: The class label (hateful, offensive, normal) of

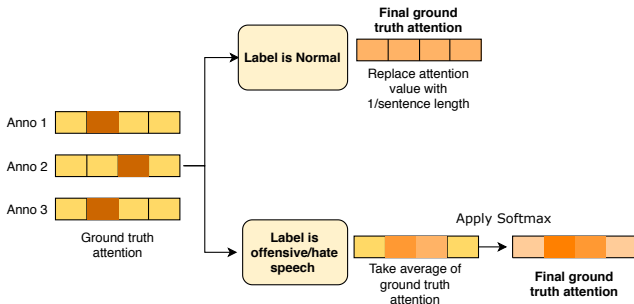


Figure 1: Ground truth attention.

a post was decided based on majority voting. We found 919 cases where all the three annotators chose a different class. We did not consider these posts for our analysis.

To decide the target community of a post, we rely on majority voting. We consider that a target community is present in the post, if at least two out of the three annotators have selected the target from Table 3. We also add a filter that the community should be present in at least 100 posts. Based on this criteria, our dataset had the following ten communities: *African, Islam, Jewish, LGBTQ, Women, Refugee, Arab, Caucasian, Hispanic, Asian*. The target community information would allow researchers to delve into issues related to bias in hate speech (Davidson, Bhattacharya, and Weber 2019). In our dataset, the top three communities that are targets of hate speech are the *African, Islam, and Jewish* community. In case of offensive speech, the top three targets are *Women, Africans, and LGBTQ*. These observations are in agreement with previous research (Silva et al. 2016).

For the rationales’ annotation, each post that is labelled as hateful or offensive was further provided to the annotators⁶ to highlight the rationales that could justify the final class. Each post had rationale explanations provided by 2-3 annotators. We observe that the average number of tokens highlighted per post is 5.48 for offensive speech, and 5.47 for hate speech. Average token per post in the whole dataset is 23.42. The top three content words in the hate speech rationales are *nigger, kike, and moslems*, which are found in 30.02% of all the hateful posts. The top three content words for the offensive highlights are *retarded, bitch, and white*, which are found in 47.36% of all the offensive posts.

Ground truth attention: In order to generate the ground truth attention for the post with hate speech/offensive label, we first convert each rationale into an attention vector. This is a Boolean vector with length equal to the number of tokens in the sentence. The tokens in the rationale are indicated by a value of 1 in the attention vector. Now we take the average of these attention vectors to represent a common ground truth attention vector for each post. The attention vectors from the attention based models usually have their sum of elements equal to 1. We normalize this common attention vector through a softmax function to generate the ground truth attention. One issue with the ground truth

⁶We tried to get the original annotator to highlight, however this was not always possible.

Text	Dad should have told the muzrat whore to fuck off , and went in anyway
Label	Hate
Targets	Islam
Text	A nigress too dumb to fuck has a scant chance of understanding anything beyond the size of a dick
Label	Hate
Targets	Women, African
Text	Twitter is full of tween dikes who think they’re superior because of ”muh oppression.” News flash: No one gives a shit.
Label	Offensive
Targets	LGBTQ

Table 5: Examples from our dataset. The **highlighted** portion of the text represents the annotator’s rationale.

attention vector could be that the difference between the values of rationale and non-rationale tokens could be low. To handle this, we make use of the temperature parameter (τ) in the softmax function. This allows us to make the probability distribution concentrate on the rationales. We tune this parameter using the validation set. Finally, if the label of the post is normal, we ignore the attention vectors and replace each element in the ground truth attention with $1/(\text{sentence length})$ to represent uniform distribution. We illustrate this computation in Figure 1.

Metrics for evaluation

To build the **HateXplain** benchmark dataset, we consider multiple types of metrics to cover several aspects of hate speech. Taking inspiration from the different issues reported for hate speech classifications, we concentrate on three major types of metrics.

Performance based metrics

Following the standard practices, we report **accuracy**, **macro F1-score**, and **AUROC** score. These metrics would be able to evaluate the classifier performance in distinguishing among the three classes, i.e., hate speech, offensive speech, and normal.

Bias based metrics

The hate speech detection models could make biased predictions for particular groups who are already the target of such abuse (Sap et al. 2019; Davidson, Bhattacharya, and Weber 2019). For example, the sentence “I love my niggas.” might be classified as hateful/offensive because of the association of the word niggas with the black community. These unintended identity-based bias could have negative impact on the target community. To measure such unintended model bias, we rely on the AUC based metrics developed by Borkan et al. (2019). These include Subgroup AUC, Background Positive Subgroup Negative (BPSN) AUC, Background Negative Subgroup Positive (BNSP) AUC, Generalized Mean of Bias AUCs. The task here is to classify the

post as *toxic* (*hate speech*, *offensive*) or *not* (*normal*). Here, the models will be evaluated on the grounds of how much they are able to reduce the unintended bias towards a target community (Borkan et al. 2019). We restrict the evaluation to the test set only. By having this restriction, we are able to evaluate models in terms of bias reduction. Below, we briefly describe each of the metrics.

Subgroup AUC: Here, we select toxic and normal posts from the test set that mention the community under consideration. The ROC-AUC score of this set will provide us with the Subgroup AUC for a community. This metric measures the model’s ability to separate the toxic and normal comments in the context of the community (e.g., Asians, LGBTQ etc.). A higher value means that the model is doing a good job at distinguishing the toxic and normal posts specific to the community.

BPSN (Background Positive, Subgroup Negative) AUC: Here, we select normal posts that mention the community and toxic posts that do not mention the community, from the test set. The ROC-AUC score of this set will provide us with the BPSN AUC for a community. This metric measures the false-positive rates of the model with respect to a community. A higher value means that a model is less likely to confuse between the normal post that mentions the community with a toxic post that does not.

BNSP (Background Negative, Subgroup Positive) AUC: Here, we select toxic posts that mention the community and normal posts that do not mention the community, from the test set. The ROC-AUC score for this set will provide us with the BNSP AUC for a community. The metric measures the false-negative rates of the model with respect to a community. A higher value means that the model is less likely to confuse between a toxic post that mentions the community with a normal post without one.

GMB (Generalized Mean of Bias) AUC: This metric was introduced by the Google Conversation AI Team as part of their Kaggle competition⁷. This metric combines the per-identity Bias AUCs into one overall measure as $M_p(m_s) = \left(\frac{1}{N} \sum_{s=1}^N m_s^p\right)^{\frac{1}{p}}$ where, M_p = the p^{th} power-mean function, m_s = the bias metric m calculated for subgroup s and N = number of identity subgroups (10). We use $p = -5$ as was also done in the competition.

We report the following three metrics for our dataset.

- **GMB-Subgroup-AUC:** GMB AUC with Subgroup AUC as the bias metric.
- **GMB-BPSN-AUC:** GMB AUC with BPSN AUC as the bias metric.
- **GMB-BNSP-AUC:** GMB AUC with BNSP AUC as the bias metric.

Explainability based metrics

We follow the framework in the ERASER benchmark by DeYoung et al. (2020) to measure the explainability aspect of a model. We measure this using *plausibility* and

faithfulness. *Plausibility* refers to how convincing the interpretation is to humans, while *faithfulness* refers to how accurately it reflects the true reasoning process of the model (Jacovi and Goldberg 2020).

For completeness, we explain the metrics briefly below.

Plausibility To measure the plausibility, we consider metrics for both discrete and soft selection. We report the IOU F1-Score and token F1-Score metric for the discrete case, and the AUPRC score for soft token selection (DeYoung et al. 2020).

Intersection-Over-Union (IOU) permits credit assignment for partial matches. DeYoung et al. (2020) defines IOU on a token level: for two spans, it is the size of the overlap of the tokens they cover divided by the size of their union. A prediction is considered as a match if the overlap with any of the ground truth rationales is more than 0.5. We use these partial matches to calculate an F1-score (IOU F1). We also measure token-level precision and recall, and use these to derive token-level F1 scores (token F1). To measure the plausibility for soft token scoring, we also report the Area Under the Precision-Recall curve (AUPRC) constructed by sweeping a threshold over the token scores.

Faithfulness To measure the faithfulness, we report two metrics: *comprehensiveness* and *sufficiency* (DeYoung et al. 2020).

- **Comprehensiveness:** To measure comprehensiveness, we create a contrast example \tilde{x}_i , for each post x_i , where \tilde{x}_i is calculated by removing the predicted rationales r_i ⁸ from x_i . Let $m(x_i)_j$ be the original prediction probability provided by a model m for the predicted class j . Then we define $m(x_i \setminus r_i)_j$ as the predicted probability of \tilde{x}_i ($= x_i \setminus r_i$) by the model m for the class j . We would expect the model prediction to be lower on removing the rationales. We can measure this as follows – comprehensiveness = $m(x_i)_j - m(x_i \setminus r_i)_j$. A high value of comprehensiveness implies that the rationales were influential in the prediction.
- **Sufficiency** measures the degree to which extracted rationales are adequate for a model to make a prediction. We can measure this as follows – sufficiency = $m(x_i)_j - m(r_i)_j$.

Model details

In this section, we provide details on the models used to evaluate the dataset. Each model has two versions, one where the models are trained using the ground truth class labels only (i.e., hate speech, offensive speech, and normal) and the other, where the models are trained using the ground truth attention and class labels, as shown in Figure 2. For training using the ground truth attention, the model needs to output some form of vector representing attention for each token according to the model, hence, the second version is not feasible for BiRNN and CNN-GRU models⁹.

⁸We select the top 5 tokens as the rationales. The top 5 is selected as it is the average length of the annotation span in the dataset.

⁹The limitation is due to the lack of an attention mechanism

⁷<https://www.kaggle.com/c/jigsaw-unintended-bias-in-toxicity-classification/overview/evaluation>

Model [Token Method]	Performance			Bias			Explainability			Faithfulness	
	Acc.↑	Macro F1↑	AUROC↑	GMB-Sub.↑	GMB-BPSN↑	GMB-BNSP↑	IOU F1↑	Plausibility Token F1↑	AUPRC↑	Comp.↑	Suff.↓
CNN-GRU [LIME]	0.627	0.606	0.793	0.654	0.623	0.659	0.167	0.385	0.648	0.316	-0.082
BiRNN [LIME]	0.595	0.575	0.767	0.640	0.604	0.671	0.162	0.361	0.605	0.421	-0.051
BiRNN-Attn [Attn]	0.621	0.614	0.795	0.653	0.662	0.668	0.167	0.369	0.643	0.278	0.001
BiRNN-Attn [LIME]	0.621	0.614	0.795	0.653	0.662	0.668	0.162	0.386	0.650	0.308	-0.075
BiRNN-HateXplain [Attn]	0.629	0.629	0.805	0.691	0.636	0.674	0.222	0.506	0.841	0.281	0.039
BiRNN-HateXplain [LIME]	0.629	0.629	0.805	0.691	0.636	0.674	0.174	0.407	0.685	0.343	-0.075
BERT [Attn]	0.690	0.674	0.843	0.762	0.709	0.757	0.130	0.497	0.778	0.447	0.057
BERT [LIME]	0.690	0.674	0.843	0.762	0.709	0.757	0.118	0.468	0.747	0.436	0.008
BERT-HateXplain [Attn]	0.698	0.687	0.851	0.807	0.745	0.763	0.120	0.411	0.626	0.424	0.160
BERT-HateXplain [LIME]	0.698	0.687	0.851	0.807	0.745	0.763	0.112	0.452	0.722	0.500	0.004

Table 6: Model performance results. To select the tokens for explainability calculation, we used attention and LIME methods.

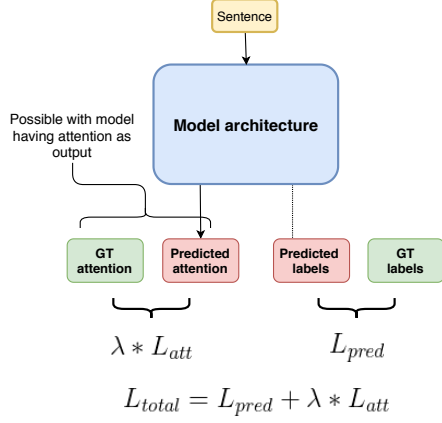


Figure 2: Representation of the general model architecture showing how the attention of the model is trained using the ground truth (GT) attention. λ controls how much effect the attention loss has on the total loss.

CNN-GRU Zhang, Robinson, and Tepper (2018) used CNN-GRU to achieve state-of-the-art for multiple hate speech datasets. We modify the original architecture to include convolution 1D filters of window sizes 2, 3, 4 with each size having 100 filters. For the RNN part, we use GRU layer and finally max-pool the output representation from the hidden layers of the GRU architecture. This hidden layer is passed through a fully connected layer to finally output the prediction logits.

BiRNN For the BiRNN (Schuster and Paliwal 1997) model, we pass the tokens in the form of embeddings to a sequential model¹⁰. The last hidden state is passed through 2 fully connected layers. The output after that is used as the prediction logits. We use dropout layers after the embedding layer and before both the fully connected layers to regularise the trained model.

BiRNN-Attention This model is identical to the BiRNN model but includes an attention layer (Liu and Lane 2016) after the sequential layer. This attention layer outputs an attention vector based on a context vector which is analogous to asking “which is the most important word?”. Weights from the attention vector are multiplied with the output hid-

den units from the sequential layer and added to present a final representation of the sentence. This representation is passed through two fully connected layers as in the BiRNN model. Further to train the attention layer outputs, we compute cross entropy loss between the attention layer output and the ground truth attention (cf. Figure 1 for its computation) as shown in Figure 2.

BERT BERT (Devlin et al. 2019) stands for Bidirectional Encoder Representations from Transformers pre-trained on data from English language¹¹. It is a stack of transformer encoder layers with 12 “attention heads”, i.e., fully connected neural networks augmented with a self attention mechanism. In order to fine-tune BERT, we add a fully connected layer with the output corresponding to the *CLS* token in the input. This *CLS* token output usually holds the representation of the sentence. Next, to add *attention supervision*, we try to match the attention values corresponding to the *CLS* token in the final layer to the ground truth attention, so that when the final weighted representation of *CLS* is generated, it would give attention to words as per the ground truth attention vector. This is calculated using a cross entropy between the attention values and the ground truth attention vector as shown in Figure 2.

Hyper-parameter tuning

All the methods are compared using the same *train:development:test* split of 8:1:1. We perform stratified split on the dataset to maintain class balance. All the results are reported on the test set and the development set is used for hyper-parameter tuning. We use the common crawl¹² pre-trained GloVe embeddings (Pennington, Socher, and Manning 2014) to initialize the word embeddings for the non-BERT models. In our models, we set the token length to 128 for faster processing of the query¹³. We use Adam (Kingma and Ba 2015) optimizer and find the learning rate to 0.001 for the non-BERT models and 2e-5 for BERT models using the development set. The RNN models prefer LSTM as the sequential layer with hidden layer size of 64 for BiRNN with attention and 128 for BiRNN. We use dropouts at different levels of the model. The regulariser λ

¹¹We use the bert-base-uncased model having 12-layer, 768-hidden, 12-heads, 110M parameters.

¹²840B tokens, 2.2M vocab, cased, 300d vectors.

¹³Almost all the posts consist of less than 128 tokens in the data.

¹⁰We experiment with LSTM and GRU.

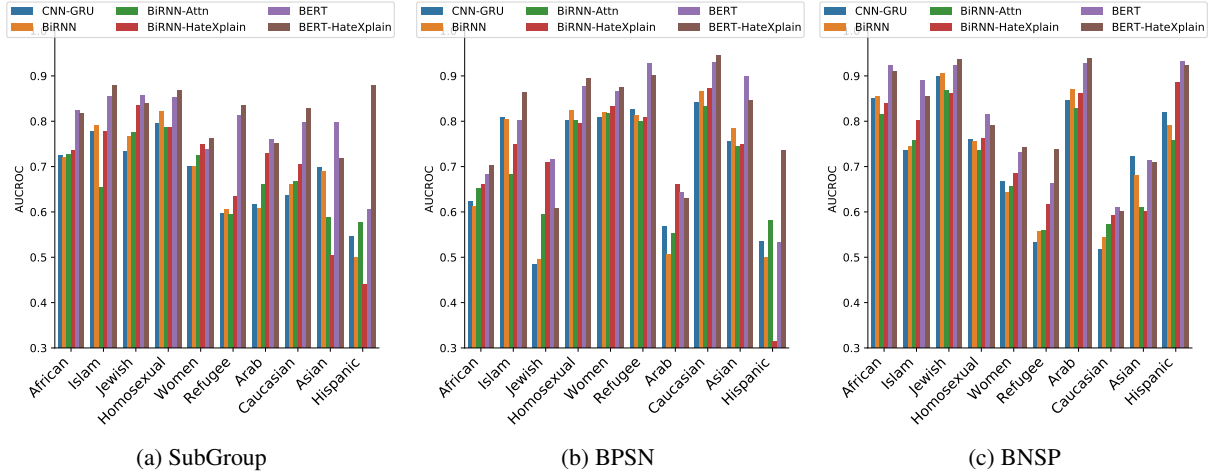


Figure 3: Community-wise results for each of the bias metrics.

controls how much effect the attention loss has on the total loss as in Figure 2. Optimum performance occurs with λ being set to 100 for BiRNN with attention and BERT with attention in the supervised setting¹⁴.

Results

We report the main results obtained in Table 6.

Performance: We observe that models utilizing the human rationales as part of the training (BiRNN-HateXplain [LIME & Attn], BERT-HateXplain [LIME & Attn])¹⁵ are able to perform slightly better in terms of the performance metrics. BiRNN-HateXplain [LIME & Attn] has improved score for all plausibility metrics and comprehensiveness as compared to BiRNN-Attn [LIME & Attn]. In case of BERT-HateXplain [LIME], the faithfulness scores have improved as compared to other BERT models. However, the plausibility scores have decreased.

Bias: Similar to performance, models that utilize the human rationales as part of the training are able to perform better in reducing the unintended model bias for all the bias metrics. We observe that presence of community terms within the rationales is effective in reducing the unintended bias. We also looked at the model bias for each individual community in Figure 3. Figure 3a reports the community wise subgroup AUROC. We observe that while the GMB-Subgroup metric reports ~ 0.8 AUROC, the score for individual community has large variations. Target communities like Asians have scores ~ 0.7 , even for the best model. Communities like Hispanic seem to be biased toward having more false positives. Models like BERT-HateXplain seem to be able to

handle this bias much better than other models. Future research on hate speech, should consider the impact of the model performance on individual communities to have a clear understanding on the impact.

Explainability: We observe that models such as BERT-HateXplain [LIME & Attn], which attain the best scores in terms of performance metrics and bias, do not perform well in terms of plausibility explainability metrics. In fact, BERT-HateXplain [Attn] has the worst score for sufficiency as compared to other models. BERT-HateXplain [LIME] seems to be the best model for comprehensiveness metric. For plausibility metrics, we observe BiRNN-HateXplain [Attn] to have the best scores. For sufficiency, CNN-GRU seems to be doing the best. For the token method, LIME seems to be generating more faithful results as compared to attention. These are in agreement with DeYoung et al. (2020). Overall, we observe that a model’s performance metric alone is not enough. Models with slightly lower performance, but much higher scores for plausibility and faithfulness might be preferred depending on the task at hand. The HateXplain dataset could be a valuable tool for researchers to analyze and develop models that provide more explainable results.

Variations with λ : We measure the effect of λ on model performance (macro F1 and AUROC) and explainability (token F1, AUPRC, comp., and suff.). We experiment with BiRNN-HateXplain [Attn] and BERT-HateXplain [Attn]. Increasing the value of λ improves the model performance, plausibility, and sufficiency while degrading comprehensiveness.

Limitations of our work

Our work has several limitations. First is the lack of external context. In our current models, we have not considered any external context such as profile bio, user gender, history of posts etc., which might be helpful in the classification task. Also, in this work we have focused on the English language. It does not consider multilingual hate speech into account.

¹⁴Please note that our selection of the best hyper-parameter was based on the model performance, which is in lines with what is suggested in the literature. One could have a variant where the model is optimized for the best explainability. This dataset gives researchers the flexibility to choose best parameters based on plausibility and/or faithfulness.

¹⁵<model>-HateXplain denotes the models where we use supervised attention using ground truth attention vector.

Conclusion and future work

In this paper, we have introduced **HateXplain**, a new benchmark dataset¹ for hate speech detection. The dataset consists of 20K posts from Gab and Twitter. Each data point is annotated with one of the hate/offensive/normal labels, target communities mentioned, and snippets (rationales) of the text marked by the annotators who support the label. We test several state-of-the-art models on this dataset and perform evaluation on several aspects of the hate speech detection. Models that perform very well in classification cannot always provide plausible and faithful rationales for their decisions.

As part of the future work, we plan to incorporate existing hate speech datasets (Davidson et al. 2017; Ousidhoum et al. 2019; Founta et al. 2018) to our **HateXplain** framework.

References

- Aluru, S. S.; Mathew, B.; Saha, P.; and Mukherjee, A. 2020. Deep Learning Models for Multilingual Hate Speech Detection. *arXiv preprint arXiv:2004.06465*.
- Arango, A.; Pérez, J.; and Poblete, B. 2019. Hate Speech Detection is Not as Easy as You May Think: A Closer Look at Model Validation. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, 45–54. Paris, France: Association for Computing Machinery.
- Basile, V.; Bosco, C.; Fersini, E.; Nozza, D.; Patti, V.; Rangel Pardo, F. M.; Rosso, P.; and Sanguinetti, M. 2019. SemEval-2019 Task 5: Multilingual Detection of Hate Speech Against Immigrants and Women in Twitter. In *Proceedings of the 13th International Workshop on Semantic Evaluation*, 54–63. Minneapolis, Minnesota, USA: Association for Computational Linguistics.
- Borkan, D.; Dixon, L.; Sorensen, J.; Thain, N.; and Vasserman, L. 2019. Nuanced Metrics for Measuring Unintended Bias with Real Data for Text Classification. In *Companion of The 2019 World Wide Web Conference, WWW*, 491–500. San Francisco, CA, USA: Association for Computing Machinery.
- Bosco, C.; Felice, D.; Poletto, F.; Sanguinetti, M.; and Maurizio, T. 2018. Overview of the EVALITA 2018 Hate Speech Detection Task. In *EVALITA 2018-Sixth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian*, volume 2263, 1–9. Turin, Italy: CEUR-WS.org.
- Burnap, P.; and Williams, M. L. 2016. Us and them: identifying cyber hate on Twitter across multiple protected characteristics. *EPJ Data Sci.* 5(1): 11.
- Camburu, O.; Rocktäschel, T.; Lukasiewicz, T.; and Blunsom, P. 2018. e-SNLI: Natural Language Inference with Natural Language Explanations. In *31st Annual Conference on Neural Information Processing Systems*, 9560–9572. Montréal, Canada: Advances in Neural Information Processing Systems.
- Council, E. 2016. EU Regulation 2016/679 General Data Protection Regulation (GDPR). *Official Journal of the European Union* 59(6): 1–88.
- Davidson, T.; Bhattacharya, D.; and Weber, I. 2019. Racial Bias in Hate Speech and Abusive Language Detection Datasets. In *Proceedings of the Third Workshop on Abusive Language Online*, 25–35. Florence, Italy: Association for Computational Linguistics.
- Davidson, T.; Warmley, D.; Macy, M. W.; and Weber, I. 2017. Automated Hate Speech Detection and the Problem of Offensive Language. In *Proceedings of the Eleventh International Conference on Web and Social Media*, 512–515. Montréal, Québec, Canada: AAAI Press.
- de Gibert, O.; Perez, N.; García-Pablos, A.; and Cuadros, M. 2018. Hate Speech Dataset from a White Supremacy Forum. In *Proceedings of the 2nd Workshop on Abusive Language Online (ALW2)*, 11–20. Brussels, Belgium: Association for Computational Linguistics.
- Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 4171–4186. Minneapolis, Minnesota: Association for Computational Linguistics.
- DeYoung, J.; Jain, S.; Rajani, N. F.; Lehman, E.; Xiong, C.; Socher, R.; and Wallace, B. C. 2020. ERASER: A Benchmark to Evaluate Rationalized NLP Models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 4443–4458. Online: Association for Computational Linguistics.
- Dixon, L.; Li, J.; Sorensen, J.; Thain, N.; and Vasserman, L. 2018. Measuring and Mitigating Unintended Bias in Text Classification. In *Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society*, 67–73. Association for Computing Machinery.
- Doshi-Velez, Finale; Kim, B. 2017. Towards A Rigorous Science of Interpretable Machine Learning. In *eprint arXiv:1702.08608*.
- Fortuna, P.; and Nunes, S. 2018. A survey on automatic detection of hate speech in text. *ACM Computing Surveys (CSUR)* 51(4): 85.
- Founta, A.; Djouvas, C.; Chatzakou, D.; Leontiadis, I.; Blackburn, J.; Stringhini, G.; Vakali, A.; Sirivianos, M.; and Kourtellis, N. 2018. Large Scale Crowdsourcing and Characterization of Twitter Abusive Behavior. In *Proceedings of the Twelfth International Conference on Web and Social Media*, 491–500. Stanford, California, USA: AAAI Press.
- Goodfellow, I.; Bengio, Y.; and Courville, A. 2016. *Deep Learning*. MIT Press. <http://www.deeplearningbook.org>.
- Greenberg, J.; and Pyszczynski, T. 1985. The effect of an overheard ethnic slur on evaluations of the target: How to spread a social disease. *Journal of Experimental Social Psychology* 21(1): 61–72.
- Gröndahl, T.; Pajola, L.; Juuti, M.; Conti, M.; and Asokan, N. 2018. All You Need is: Evading Hate Speech Detection. In *Proceedings of the 11th ACM Workshop on Artificial Intelligence and Security*, 2–12. Toronto, ON, Canada: Association for Computing Machinery.

- Jacovi, A.; and Goldberg, Y. 2020. Towards Faithfully Interpretable NLP Systems: How Should We Define and Evaluate Faithfulness? In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 4198–4205. Online: Association for Computational Linguistics.
- Kingma, D. P.; and Ba, J. 2015. Adam: A Method for Stochastic Optimization. In *Proceedings of 3rd International Conference on Learning Representations*. San Diego, CA, USA: International Conference on Learning Representations.
- Lei, T.; Barzilay, R.; and Jaakkola, T. S. 2016. Rationalizing Neural Predictions. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, 107–117. Austin, Texas, USA: The Association for Computational Linguistics.
- Lima, L.; Reis, J. C.; Melo, P.; Murai, F.; Araujo, L.; Vikatos, P.; and Benevenuto, F. 2018. Inside the right-leaning echo chambers: Characterizing gab, an unmoderated social system. In *2018 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*, 515–522. Barcelona, Spain: IEEE Computer Society.
- Lipton, Z. C. 2018. The mythos of model interpretability. *Communications of the ACM* 61(10): 36–43.
- Liu, B.; and Lane, I. 2016. Attention-Based Recurrent Neural Network Models for Joint Intent Detection and Slot Filling. In *Interspeech 2016, 17th Annual Conference of the International Speech Communication Association*, 685–689. San Francisco, CA, USA: International Speech Communication Association.
- Mathew, B.; Dutt, R.; Goyal, P.; and Mukherjee, A. 2019a. Spread of hate speech in online social media. In *Proceedings of the 10th ACM Conference on Web Science*, 173–182. Boston, MA, USA: Association for Computing Machinery.
- Mathew, B.; Illendula, A.; Saha, P.; Sarkar, S.; Goyal, P.; and Mukherjee, A. 2020. Hate Begets Hate: A Temporal Study of Hate Speech. *Proceedings of the ACM on Human-Computer Interaction*.
- Mathew, B.; Saha, P.; Tharad, H.; Rajgaria, S.; Singhanian, P.; Maity, S. K.; Goyal, P.; and Mukherjee, A. 2019b. Thou shalt not hate: Countering online hate speech. In *Proceedings of the International AAAI Conference on Web and Social Media*, 369–380. Munich, Germany: AAAI Press.
- Mishra, P.; Del Tredici, M.; Yannakoudakis, H.; and Shutova, E. 2018. Author profiling for abuse detection. In *Proceedings of the 27th International Conference on Computational Linguistics*, 1088–1098. Santa Fe, New Mexico, USA: Association for Computational Linguistics.
- Olteanu, A.; Castillo, C.; Boy, J.; and Varshney, K. R. 2018. The Effect of Extremist Violence on Hateful Speech Online. In *Proceedings of the Twelfth International Conference on Web and Social Media*, 221–230. Stanford, California, USA: AAAI Press.
- Ousidhoum, N.; Lin, Z.; Zhang, H.; Song, Y.; and Yeung, D.-Y. 2019. Multilingual and Multi-Aspect Hate Speech Analysis. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 4667–4676. Hong Kong, China: Association for Computational Linguistics.
- Pennington, J.; Socher, R.; and Manning, C. D. 2014. Glove: Global Vectors for Word Representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, 1532–1543. Doha, Qatar: Association for Computational Linguistics.
- Qian, J.; Bethke, A.; Liu, Y.; Belding, E. M.; and Wang, W. Y. 2019a. A Benchmark Dataset for Learning to Intervene in Online Hate Speech. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, 4754–4763. Hong Kong, China: Association for Computational Linguistics.
- Qian, J.; ElSherief, M.; Belding, E. M.; and Wang, W. Y. 2018a. Hierarchical CVAE for Fine-Grained Hate Speech Classification. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 3550–3559. Brussels, Belgium: Association for Computational Linguistics.
- Qian, J.; ElSherief, M.; Belding, E. M.; and Wang, W. Y. 2018b. Leveraging Intra-User and Inter-User Representation Learning for Automated Hate Speech Detection. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 118–123. New Orleans, Louisiana, USA: Association for Computational Linguistics.
- Qian, J.; ElSherief, M.; Belding, E. M.; and Wang, W. Y. 2019b. Learning to Decipher Hate Symbols. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 3006–3015. Minneapolis, MN, USA: Association for Computational Linguistics.
- Rajani, N. F.; McCann, B.; Xiong, C.; and Socher, R. 2019. Explain Yourself! Leveraging Language Models for Commonsense Reasoning. In *Proceedings of the 57th Conference of the Association for Computational Linguistics*, 4932–4942. Florence, Italy: Association for Computational Linguistics.
- Ribeiro, M. H.; Calais, P. H.; Santos, Y. A.; Almeida, V. A. F.; and Jr., W. M. 2018. Characterizing and Detecting Hateful Users on Twitter. In *Proceedings of the Twelfth International Conference on Web and Social Media*, 676–679. Stanford, California, USA: AAAI Press.
- Ribeiro, M. T.; Singh, S.; and Guestrin, C. 2016. "Why Should I Trust You?": Explaining the Predictions of Any Classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 1135–1144. New York, NY, USA: Association for Computing Machinery.
- Sanguinetti, M.; Poletto, F.; Bosco, C.; Patti, V.; and Stranisci, M. 2018. An Italian Twitter Corpus of Hate Speech against Immigrants. In *Proceedings of the Eleventh*

International Conference on Language Resources and Evaluation. Miyazaki, Japan: European Language Resources Association.

Sap, M.; Card, D.; Gabriel, S.; Choi, Y.; and Smith, N. A. 2019. The Risk of Racial Bias in Hate Speech Detection. In *Proceedings of the 57th Conference of the Association for Computational Linguistics*, 1668–1678. Florence, Italy: Association for Computational Linguistics.

Schuster, M.; and Paliwal, K. K. 1997. Bidirectional recurrent neural networks. *IEEE Trans. Signal Process.* 45(11): 2673–2681.

Silva, L. A.; Mondal, M.; Correa, D.; Benevenuto, F.; and Weber, I. 2016. Analyzing the Targets of Hate in Online Social Media. In *Proceedings of the Tenth International Conference on Web and Social Media*, 687–690. Cologne, Germany: AAAI Press.

Soral, W.; Bilewicz, M.; and Winiewski, M. 2018. Exposure to hate speech increases prejudice through desensitization. *Aggressive behavior* 44(2): 136–146.

Van Huynh, T.; Nguyen, V. D.; Van Nguyen, K.; Nguyen, N. L.-T.; and Nguyen, A. G.-T. 2019. Hate Speech Detection on Vietnamese Social Media Text using the Bi-GRU-LSTM-CNN Model. *arXiv preprint arXiv:1911.03644*.

Vigna, F. D.; Cimino, A.; Dell’Orletta, F.; Petrocchi, M.; and Tesconi, M. 2017. Hate Me, Hate Me Not: Hate Speech Detection on Facebook. In *Proceedings of the First Italian Conference on Cybersecurity*, volume 1816, 86–95. Venice, Italy: CEUR-WS.org.

Wang, W.; Chen, L.; Thirunarayan, K.; and Sheth, A. P. 2014. Cursing in English on twitter. In *Proceedings of the 17th ACM conference on Computer supported cooperative work & social computing*, 415–425. Baltimore, MD, USA: Association for Computing Machinery.

Waseem, Z.; and Hovy, D. 2016. Hateful Symbols or Hateful People? Predictive Features for Hate Speech Detection on Twitter. In *Proceedings of the NAACL Student Research Workshop*, 88–93. San Diego, California, USA: The Association for Computational Linguistics.

Williams, M. L.; Burnap, P.; Javed, A.; Liu, H.; and Ozalp, S. 2020. Hate in the machine: anti-Black and anti-Muslim social media posts as predictors of offline racially and religiously aggravated crime. *The British Journal of Criminology* 60(1): 93–117.

Yessenalina, A.; Choi, Y.; and Cardie, C. 2010. Automatically Generating Annotator Rationales to Improve Sentiment Classification. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, 336–341. Uppsala, Sweden: The Association for Computer Linguistics.

Zaidan, O.; Eisner, J.; and Piatko, C. D. 2007. Using “Annotator Rationales” to Improve Machine Learning for Text Categorization. In *Proceedings of Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics*, 260–267. Rochester, New York, USA: The Association for Computational Linguistics.

Zannettou, S.; Bradlyn, B.; Cristofaro, E. D.; Kwak, H.; Sirivianos, M.; Stringhini, G.; and Blackburn, J. 2018. What is Gab: A Bastion of Free Speech or an Alt-Right Echo Chamber. In *Companion of the The Web Conference 2018 on The Web Conference 2018, WWW 2018, April 23-27, 2018*, 1007–1014. Lyon, France: Association for Computing Machinery.

Zhang, Z.; Robinson, D.; and Tepper, J. A. 2018. Detecting Hate Speech on Twitter Using a Convolution-GRU Based Deep Neural Network. In *Proceedings of The Semantic Web - 15th International Conference*, 745–760. Heraklion, Crete, Greece: Springer.

Appendix

Interface design

We divided the interface into two parts. First, as shown in Figure 4, we classified the text as hate speech, offensive, or normal, along with the targets in the text. Second, in Figure 5, we asked the annotators to highlight the portions of the text that could justify the label given to the text (hate speech/offensive). In order to help the annotators with the task, we provided them with multiple example annotations and highlights. We also provided them with sample test cases (as shown in Figure 6) to test out the highlight system.

Hate Speech Classification

Instructions

Please read Examples and Task Definitions Carefully. Please check the feedback that we have sent based on your first round of annotation. We will monitor your annotation to make sure that you are following the guidelines.

Search Definitions

If you do not know the meaning of any terms in the texts below, write it here and search (opens in new window):

Search Definition Search

1. Select the appropriate option for each of the ten sentences:

#	Sentence	Classification	Race	Religion	Gender	Sexual Orientation	Others
1.	@user I wish we could just go around killing all these fucking jews.	<input checked="" type="radio"/> Hate Speech <input type="radio"/> Offensive but NOT hateful <input type="radio"/> Normal (neither hateful nor offensive)	<input type="checkbox"/> African <input type="checkbox"/> Arab <input type="checkbox"/> Asian <input type="checkbox"/> Caucasian <input type="checkbox"/> Hispanic	<input type="checkbox"/> Buddhism <input type="checkbox"/> Christian <input type="checkbox"/> Hindu <input type="checkbox"/> Islam <input checked="" type="checkbox"/> Jewish	<input type="checkbox"/> Men <input type="checkbox"/> Women	<input type="checkbox"/> Heterosexual <input type="checkbox"/> Homosexual	<input type="checkbox"/> Indigenous <input type="checkbox"/> Refugee/Immigrant <input type="checkbox"/> None (No Target Present) <input type="checkbox"/> Others:
2.	Muslims are the worst of our species. They are better off dead.	<input checked="" type="radio"/> Hate Speech <input type="radio"/> Offensive but NOT hateful <input type="radio"/> Normal (neither hateful nor offensive)	<input type="checkbox"/> African <input type="checkbox"/> Arab <input type="checkbox"/> Asian <input type="checkbox"/> Caucasian <input type="checkbox"/> Hispanic	<input type="checkbox"/> Buddhism <input type="checkbox"/> Christian <input type="checkbox"/> Hindu <input checked="" type="checkbox"/> Islam <input type="checkbox"/> Jewish	<input type="checkbox"/> Men <input type="checkbox"/> Women	<input type="checkbox"/> Heterosexual <input type="checkbox"/> Homosexual	<input type="checkbox"/> Indigenous <input type="checkbox"/> Refugee/Immigrant <input type="checkbox"/> None (No Target Present) <input type="checkbox"/> Others:

Figure 4: The classification interface. The annotator is provided with 20 text messages and asked to selected the correct type and target of the message.

1. Highlight the Hateful/Offensive parts of the posts:

#	Text	Label
1.	@user I couldn't quit laughing at it yesterday . White cucks out in their bass boats , risking their lives to save the niggers and Mexicans . Seriously , fuck those cucks , fuck the brown people , and fuck all the non-Texans who've moved in there . Oh , and fuck the nigger mayor	Hatespeech
2.	Smallest amount in the kike bowl .	Hatespeech
3.	come on cunt lez go the money and powrr out da back door cunt face dika	Offensive
4.	This what happens when little Adam is way smarter than his fat dyke lib teacher .	Offensive
5.	@user Yeah and today 1,000 illegal aliens were released in Texas !	Hatespeech

Figure 5: Rationale highlight. The annotators are asked to highlight the portions of the text that would justify the label.

Table 7: This table represents the different hyper-parameter variations that we tried while tuning this model.

Hyper-parameters	BERT	BiRNN	BiRNN-Attention	CNN-GRU
No. of hidden units in sequential layer	-NA-	64, 128	64, 128	-NA-
Sequential layers type	-NA-	LSTM,GRU	LSTM,GRU	GRU
Train embedding layer	-NA-	True, False	True, False	True, False
Dropout after embedding layer	-NA-	0.1,0.2,0.5	0.1,0.2,0.5	0.1,0.2,0.5
Dropout after fully connected layer	0.1,0.2,0.5	0.1,0.2,0.5	0.1,0.2,0.5	0.1,0.2,0.5
Learning rate	2e-4,2e-5,2e-6	0.1,0.01,0.001	0.1,0.01,0.001	0.1,0.01,0.001
For supervised part				
Attention lambda (λ)	0.001,0.01,0.1, 1,10,100	-NA-	0.001,0.01,0.1, 1,10,100	-NA-
Number of supervised heads (x)	1,6,12	-NA-	-NA-	-NA-

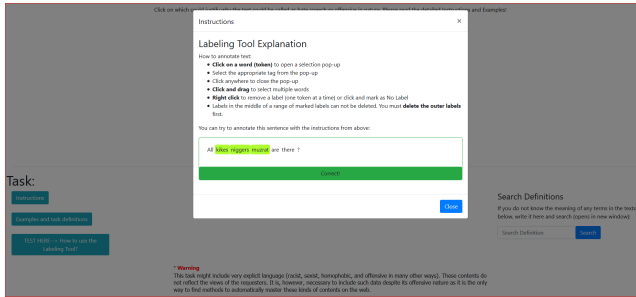


Figure 6: Highlight testing. The annotators are provided further instructions on how to highlight the texts.

Attention supervision in BERT

In each encoder layer of BERT, an attention head computes key and query vectors to generate the *attention values* for each token, based on other tokens in the sequence. These attention values multiplied with the input token representations generates the weighted encoded representation of the token. This way we get a representation of each token from each of the 12 heads. The outputs of each head in the same layer are combined and run through a fully connected layer. Each layer is wrapped with a skip-connection and a layer normalization is applied after it.

For attention supervision, we use x heads out of 12 heads in the last layer of BERT. We call these heads — *supervised heads*. For each supervised head, we use the attention weights corresponding to $[CLS]^{16}$ and calculate the cross entropy loss with ground truth attention vector as shown in Figure 7. This ensures that the final weighted output corresponding to CLS will give attention to words similar to the ground truth attention vector. Similarly, we do these steps for all *supervised heads*. The final loss from the attention supervision is the average of the cross entropy loss from each supervised heads, which is further multiplied with the regulariser- λ . Other details about the finetuning is noted in

¹⁶the first row in the $m * m$ attention weight matrix where m is the number of tokens in the tokenized sentence.

BERT section of the main paper.

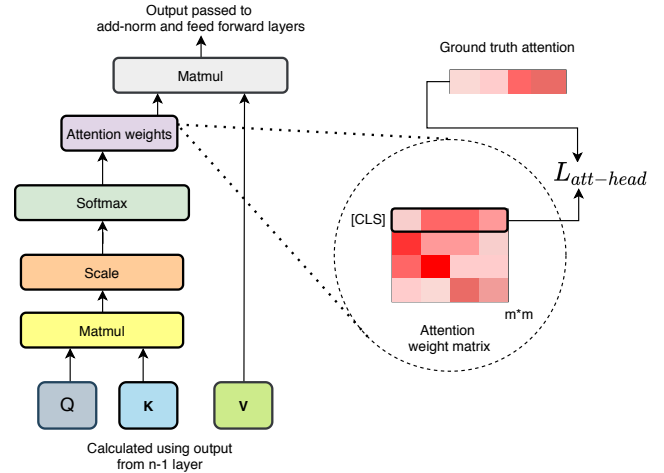


Figure 7: In this figure, we show attention supervision for a particular head in the n^{th} layer. For this work, we use the last layer for attention supervision. The number of heads for training is a hyper-parameter in our experiment.

Hyper-parameters used

Table 7 reports the hyper-parameters we tested for our systems.

More examples

Table 8 lists more examples corresponding to model predictions and rationales. In this example, BERT-HateXplain which uses token level rationales can attend better as compared to BERT. The prediction outcome is also correct for the BERT-HateXplain model. Wrong/incomplete attention (as shown) is one of the reason for incorrect predictions in BERT. For many of the false positives in BERT-HateXplain, the attended words are correct. In future, we plan to devise better mechanisms that can utilise attention for prediction.

Model	Text	Label
Human Annotator	I hate arabs	HS
BERT	I hate arabs	Normal
BERT-HateXplain	I hate arabs	HS

Table 8: Example of the rationales predicted by different models compared to human annotators. The green highlight marks tokens that the human annotator and the model finds important for the prediction.

Annotations

We looked into the quality of the fine-grained annotations as well. For this we computed the average pairwise Jaccard overlap between ground truth rationale annotations and compared them with the average pairwise overlap obtained through random annotation of the rationales. To generate the random rationale annotation, we chose 5 random tokens from a sentence as the rationale. For each sentence, we repeat this trial three times in order to denote 3 annotators. The average pairwise Jaccard overlap between the ground-truth annotations is 0.54, as compared to the random baseline of 0.36. This means that the annotators had more agreement on the token span annotations.