Rule By Example: Harnessing Logical Rules for Explainable Hate Speech Detection

Christopher Clarke[*†] **Matthew Hall**[‡] **Gaurav Mittal**[‡] **Ye Yu**[‡]
Sandra Sajeev[‡] **Jason Mars**[†] **Mei Chen**[‡]

[†]University of Michigan, Ann Arbor, MI
[‡]Microsoft, Redmond, WA
{csclarke, profmars}@umich.edu
{mathall, gaurav.mittal, yu.ye, ssajeev, mei.chen}@microsoft.com

Abstract

Classic approaches to content moderation typically apply a rule-based heuristic approach to flag content. While rules are easily customizable and intuitive for humans to interpret, they are inherently fragile and lack the flexibility or robustness needed to moderate the vast amount of undesirable content found online today. Recent advances in deep learning have demonstrated the promise of using highly effective deep neural models to overcome these challenges. However, despite the improved performance, these data-driven models lack transparency and explainability, often leading to mistrust from everyday users and a lack of adoption by many platforms. In this paper, we present **Rule By Example** (RBE): a novel exemplar-based contrastive learning approach for learning from logical rules for the task of textual content moderation. RBE is capable of providing rule-grounded predictions, allowing for more explainable and customizable predictions compared to typical deep learning-based approaches. We demonstrate that our approach is capable of learning rich rule embedding representations using only a few data examples. Experimental results on 3 popular hate speech classification datasets show that RBE is able to outperform state-of-the-art deep learning classifiers as well as the use of rules in both supervised and unsupervised settings while providing explainable model predictions via rule-grounding.

1 Introduction

Content moderation is a major challenge confronting the safety of online social platforms such as Facebook, Twitter, YouTube, Twitch, etc. (Vaidya et al., 2021). Major technology corporations are increasingly allocating valuable resources towards the development of automated systems for the detection and moderation of harmful content in addition to hiring and training expert human moderators to combat the growing menace of negativity and toxicity online (Wagner and Bloomberg, 2021; Liu et al., 2022).

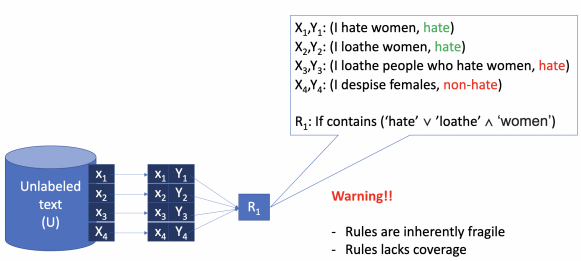

Figure 1: Generalization problem of rules. Logical rules, while easy to explain, are inherently fragile to the nuances of natural language.

Despite the popularity of deep learning approaches, many practical solutions used in products today are comprised of rule-based techniques based on expertly curated signals such as block lists, key phrases, and regular expressions (Gillespie, 2018; Zhang, 2019; Dada et al., 2019). Such methods are widely used due to their transparency, ease of customization, and interpretability. However, they have the disadvantage of being difficult to maintain and scale, in addition to being inherently fragile and noisy (Zhang, 2019; Davidson et al., 2017; Lee, 2022; Lai et al., 2022). Figure 1 shows an example where logical rules, while explainable in nature, face the problem of being inflexible to their context of use in natural language. While a given rule may be too specific and fail to capture different variations of usage commonly found in content online, rules can also be too broad and incorrectly block lexically similar content.

In contrast to the challenges faced by rule-based methods, data-driven deep learning approaches have shown great promise across a wide range of content moderation tasks and modalities (Malik et al., 2022; Shido et al., 2022; Lai et al., 2022). Fueled by large amounts of data and deep neural networks, these complex models are capable of

[*]This work was done as Christopher's internship project at Microsoft.

Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics
Volume 1: Long Papers, pages 364–376

learning richer representations that better generalize to unseen data. The impressive performances of these models have resulted in significant industry investment in content moderation as-a-service. Several technology companies such as Google [1], OpenAI [2], and Microsoft [3] use these models to offer services to aid in content moderation. However, despite their significant investment, they face adoption challenges due to the inability of customers to understand how these complex models reason about their decisions (Tarasov, 2021; Haimson et al., 2021; Juneja et al., 2020). Additionally, with the increasing attention around online content moderation and distrust amongst consumers, explainability and transparency are at the forefront of demands (Kemp and Ekins, 2021; Mukherjee et al., 2022). This presents the challenging open question of how we can leverage the robustness and predictive performance of complex deep-learning models whilst allowing the transparency, customizability, and interpretability that rule-based approaches provide.

Prior works such as Awasthi et al. (2020); Seo et al. (2021); Pryzant et al. (2022) have explored learning from rules for tasks such as controlling neural network learning, assisting in human annotation, and improving self-supervised learning in low data scenarios. Awasthi et al. (2020) propose a rule-exemplar training method for noisy supervision using rules. While performant in denoising over-generalized rules in the network via a soft implication loss, similar to other ML approaches, this method lacks the ability to interpret model predictions at inference time. Pryzant et al. (2022) propose a general-purpose framework for the automatic discovery and integration of symbolic rules into pre-trained models. However, these symbolic rules are derived from low-capacity ML models on a reduced feature space. While less complex than large deep neural networks, these low-capacity models are still not easily interpretable by humans. Therefore, the task of combining the explainability of rules and the predictive power of deep learning models remains an open problem.

In order to tackle this problem, we introduce **Rule By Example** (RBE): a novel exemplar-based contrastive learning approach for learning from logical rules for the task of textual content moderation. RBE is comprised of two neural networks, a rule encoder, and a text encoder, which jointly learn rich embedding representations for hateful content and the logical rules that govern them. Through the use of contrastive learning, our framework uses a semantic similarity objective that pairs hateful examples with clusters of rule exemplars that govern it. Through this approach, RBE is able to provide more explainable predictions by allowing for what we define as *Rule-grounding*. This means that our model is able to ground its predictions by showing the corresponding explainable logical rule and the exemplars that constitute that rule.

We evaluate RBE in both supervised and unsupervised settings using a suite of rulesets. Our results show that with as little as one exemplar per rule, RBE is capable of outperforming state-of-the-art hateful text classifiers across three benchmark content moderation datasets in both settings. In summary, the contributions of this paper are:

- Rule By Example (RBE): a novel exemplar-based contrastive learning approach to learn from logical rules for the task of textual content moderation.[4]
- We demonstrate how RBE can be easily integrated to boost model F1-score by up to 4% on three popular hate speech classification datasets.
- A detailed analysis and insights into the customizability and interpretability features of RBE to address the problem of emerging hateful content and model transparency.

2 Rule By Example Framework

In this section, we outline the Rule By Example framework, define its operational terms, and describe its end-to-end architecture. We first formally describe the two main operational terms used in our framework: 1) **Ruleset** - a ruleset is comprised of a series of executable functions that when given text as input "fire" if and only if all conditions defined in the rule are met by the input. Figure 1 shows an example of a simple rule that is triggered if a given text contains the keywords *"hate"* or

[1] https://perspectiveapi.com/

[2] https://openai.com/blog/new-and-improved-content-moderation%2Dtooling/

[3] https://azure.microsoft.com/en-us/products/cognitive-services/content-moderator/

[4] https://github.com/ChrisIsKing/Rule-By-Example

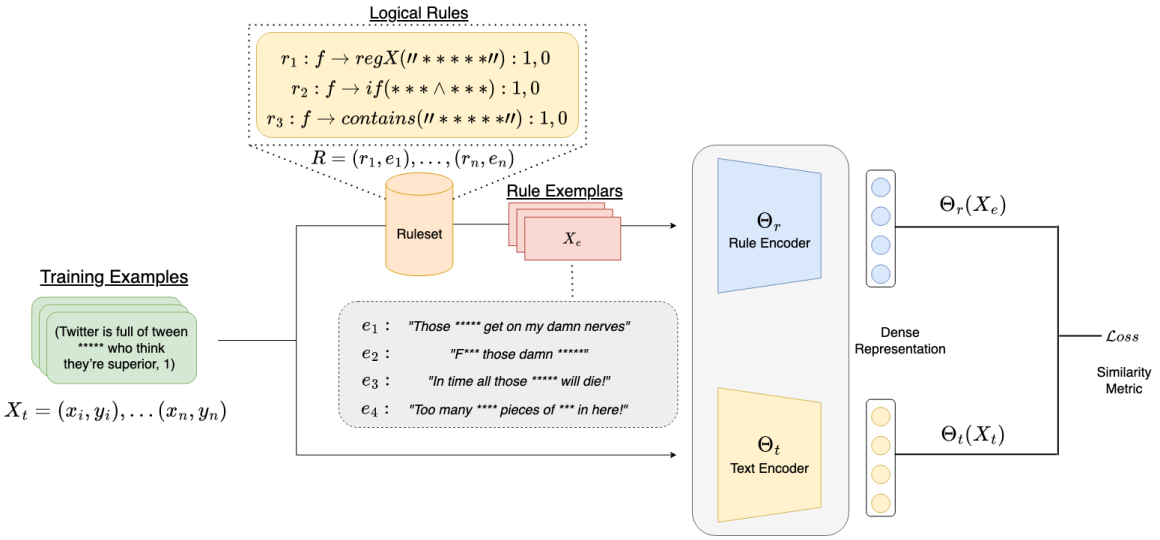

Figure 2: Rule By Example Framework: RBE is comprised of two neural networks, a rule encoder and a text encoder, which jointly learn rich embedding representations for hateful content and the logical rules that govern them. Through Contrastive learning, RBE utilizes a semantic similarity objective that pairs hateful examples with clusters of rule exemplars that govern it.

"loathe" and contains *"women"*. Rules can be any programmable function that acts on text such as regular expressions, blocklists, keywords, etc. In the scope of this work, we only consider simple rules that humans can easily interpret. As such an ML model cannot be considered a rule, given their black-box nature. 2) **Exemplar** - an exemplar is a given textual example that well-defines the type of content governed by a rule. For example, X_1 and X_2 in Figure 1 can be considered exemplars of rule R_1 since they correctly match the conditions of R_1.

Consider a ruleset of rule-exemplar pairs $R{=}\{(r_1, e_1), (r_2, e_2), ..., (r_n, e_n)\}$ where r_i denotes a defined rule and e_i denotes an exemplar for which r_i correctly fires. For a given corpus X comprising labeled examples $X{=}\{(x_1, y_1), (x_2, y_2), ..., (x_m, y_m)\}$, each rule r_i can be used as a black-box function $R_i : x \rightarrow \{y_i, \varnothing\}$ to noisily label each instance x such that it assigns a label y or no label at all. An instance may be covered by more than one rule or no rule at all. Additionally, the cover set C denotes the set of instances in X where a rule r_i fires. The generalization problem that arises when rules are applied noisily is two-fold. When rules are too broad the cover set C is large and incorrectly labels a large amount of non-hateful content. Likewise, when rules are too strict and fragile, the cover set C is too small, and lexically and semantically similar content that is hateful ends up being ignored. Our goal is to leverage these rules and their exemplars to facilitate explainable model learning.

Algorithm 1 Supervised Dual Encoder Training

Require: Rule Encoder Θ_r Text Encoder Θ_t
Input: Training Data $X{=}(x_1, y_1)...(x_n, y_n)$, Ruleset $R{=}(r_1, e_1), ..., (r_n, e_n)$
Output: Updated parameters Θ_r, Θ_t

```
Initialize Θ_r and Θ_t
while not converged do
    Get mini-batch X_b
    for each instance x_i in X_b do
        Get exemplars e_i = doRuleset(R, x_i)
        Concatenate exemplars e_i
    end for
    Get Θ_r(E_b) and Θ_t(X_b)
    Compute L = ½(Y_b D² + (1 − Y_b) max(margin − D, 0)²)
    Update parameters of Θ_r and Θ_t
end while
```

2.1 Dual Encoder Architecture

The Dual-Encoder architecture, as illustrated in Figure 2, is commonly used in dense retrieval systems and multi-modal applications (Clarke et al., 2022; Reimers and Gurevych, 2019; Xu et al., 2022). Our architecture consists of a Rule Encoder Θ_r and a Text Encoder Θ_t. These are two Bert-like bidirectional transformer models (Devlin et al., 2018) each responsible for learning embedding representations of their respective inputs. This Dual Encoder architecture enables pre-indexing of exemplars allowing for faster inference at runtime after training.

Encoding Pipeline Given an input text x_t, we first extract the set of applicable rules and their respective exemplars from the ruleset R. We then concatenate each extracted exemplar to form x_e. In the event that no rules are applicable to x_t, we randomly sample exemplars from the en-

tire ruleset to form x_e. Using the form $x_e = \{[CLS], e_1^1, ..., e_m^1, [SEP], e_1^n, ..., e_k^n\}$, we then use rule encoder Θ_r to encode x_e into hidden states $h_e = \{v_{[CLS]}, v_1, ..., v_{[SEP]}\}$ where e_k^n is the k-th token of the n-th exemplar and $[SEP]$ and $[CLS]$ are special tokens. Similarly, using the text encoder Θ_t, we encode x_t. In order to obtain a dense representation, we apply a mean pooling operation to the hidden states and derive a fixed-sized sentence embedding. After obtaining the representation for both the exemplars x_e and the text x_t, we use the cosine function to measure the similarity between them:

$$sim(x_e, x_t) = \frac{\Theta_r(x_e) \cdot \Theta_t(x_t)}{\|\Theta_r(x_e)\| \, \|\Theta_t(x_t)\|} \quad (1)$$

We employ a contrastive loss (Hadsell et al., 2006) to learn the embedding representations for our rule and text encoder. Contrastive learning encourages the model to maximize the representation similarity between *same-label* examples and to minimize it for *different-label* examples. This enables the embedding representations of our encoded ruleset to match the representation of the text correctly covered by cover set C. Likewise, for benign examples that rules incorrectly cover, our contrastive learning objective increases the distance between those representations, thus restricting the over-generalization of certain rules in the ruleset. Let Y_t be the correct label of the texts X_t, D be the cosine distance of (x_e, x_t) and m be the margin, our contrastive learning loss function is defined as follows:

$$\mathcal{L} = \frac{1}{2}(Y_t D^2 + (1 - Y_t) max(m - D, 0)^2) \quad (2)$$

The training loop, with the encoding pipeline and constrastive loss step, are detailed in Algorithm 1.

2.2 Rule-Grounding

By taking an embeddings-based approach to learning representations, RBE enables what we define as *rule-grounding*. Rule-grounding enables us to trace our model predictions back to the explainable ruleset accompanied by the exemplars that define each rule. For any input x_t that has been marked as positive by our dual encoder, we perform a rules search to find which rules fire on that input as well as an embedding similarity search to find the nearest exemplars and the rules those exemplars belong to. Table 2 shows an example of this.

3 Experimental Setup

Training We train all models with AdamW optimizer and weight decay of 0.01 on all data. We employ early stopping with a ceiling of 10 epochs, a learning rate of 2e-5, batch size of 8, and linear learning rate warmup over the first 10% steps with a cosine schedule. Our models are trained with NVIDIA Tesla V100 32GB GPUs using Azure Machine Learning Studio. We pre-process data and train all models with different random seeds over multiple runs. Our implementation of RBE is based on Huggingface Transformers (Wolf et al., 2020) and Sentence Transformers (Reimers and Gurevych, 2019). RBE utilizes two Bert-based networks consisting of 110 million parameters each. Approximately 2,000 GPU hours were required to train all hyperparameter variations of RBE plus the Bert baseline across all 3 test sets.

Baselines We evaluate our training algorithms in both supervised and unsupervised settings. We compare against the baselines of applying logical rules as is and the current SOTA approach of training transformer-based sequence classifiers (Mathew et al., 2020).

3.1 Datasets

We evaluate RBE across three datasets on the task of hate-speech classification. Across each dataset, we frame the problem as a binary classification task of detecting whether a given text is hateful or non-hateful. We augment each dataset with rulesets that we manually curate. More information on each dataset and ruleset is provided below.

HateXplain (Mathew et al., 2020) is a large-scale benchmark dataset for explainable hate speech detection that covers multiple aspects of hate speech detection. It consists of ∼20k samples across 3 labels "hateful", "offensive", and "normal". Additionally, each sample is accompanied by a corresponding target group and explainable rationales. In our experiments, we combine the output classes of hateful and offensive into one resulting in ∼8k/1k/1k hateful samples and ∼6k/781/782 non-hateful samples for train/validation/test respectively. Additionally, we utilize the accompanying rationales for ruleset construction.

Jigsaw[5] is a large-scale dataset of Wikipedia comments labeled by human raters for toxic behavior. The defined types of toxicity are "toxic", "severe toxic", "obscene", "threat", "insult", and "identity hate". Each comment can have any one or more of these labels. In total, it contains ∼230k samples. In our experiments, we define examples of the "identity hate" class as hateful and the rest as non-hateful resulting in a dataset of 1405/100/712 hateful samples and ∼158k/1k/63k non-hateful examples for train/validation/test respectively.

Contextual Abuse Dataset (CAD) (Vidgen et al., 2021) is annotated dataset of ∼25k Reddit entries labeled across six conceptually distinct primary categories of "Identity-directed", "Person-directed", "Affiliation directed", "Counter Speech", "Non-hateful Slurs", and "Neutral". In our experiment, we define examples of the "identity-directed" class as hateful and treat the remaining examples as non-hateful resulting in a dataset of 1353/513/428 hateful samples and ∼12k/4k/4k non-hateful samples for train/validation/test.

3.2 Ruleset Construction

Hate+Abuse List We utilize a ruleset targeting identity hate which we'll refer to as **Hate+Abuse List**. It consists of a list of n-grams representing harmful language such as slurs or hate verbs. Hate+Abuse List is similar to the publically available bad word lists commonly found online. We treat each n-gram entry in Hate+Abuse List as its own rule that proposes a positive label if the n-gram is in the input text. In total, Hate+Abuse List consists of 2957 distinct identity hate rules.

HateXplain Rationale Ruleset Using the labeled annotator rationales included in the HateXplain dataset, we programmatically generate a Ruleset for HateXplain. To do so, we extract 1, 2, and 3-gram substrings from the annotator rationales and cluster them by annotator-identified target demographic groups. We then take the top N n-grams per each demographic group and automatically create rules for each of them. This results in rules similar in nature to our Hate+Abuse List. Using a default cluster size of 100 across the 25 target categories defined in HateXplain, we generated a total of 670 distinct rules for HateXplain.

[5] https://www.kaggle.com/competitions/jigsaw-toxic-comment-classification%2Dchallenge

Contextual Abuse Rationale Ruleset Similar to our derived HateXplain ruleset we programmatically generate a Ruleset for the Contextual Abuse Dataset using annotator-labeled rationales. Following the identical process outlined before, this results in a total of 2712 distinct rules for CAD.

Exemplar Selection For each dataset we complete our Ruleset construction by pairing each rule with accompanying exemplars. To achieve this, we first run our Ruleset on the dataset trainset and extract instances for which a rule correctly fires. For each rule that correctly fires, we then randomly select N instances to act as the exemplars. Additionally, to restrict potentially overgeneralized rules we enforce the condition that no two rules can be mapped to the same exemplar. Unless stated otherwise, we report results using just one exemplar per rule in our experiments.

3.3 Unsupervised Setting

In addition to evaluating RBE in supervised settings, we investigate the applicability of RBE in unsupervised settings where no labeled data is present. In this setting, we are presented with a large unlabeled corpus T and a given ruleset R. This setting is particularly challenging due to the inherent generalization problem of rules. Loosely applying rules as is in this setting results in the model overfitting to the distribution of the ruleset as seen in Table 3. To combat this issue, we design three different semantic clustering-based strategies for determining rule quality in an unsupervised setting: *Mean*, *Concat*, and *Distance* clustering. Given an unlabeled corpus $T = \{t_1, t_2, ..., t_n\}$, ruleset $R = \{(r_1, e_1), ..., (r_n, e_n)\}$, and a threshold k, we first encode the entire corpus T using a pre-trained sentence embedding model E_Θ. In our case, we use a fine-tuned version of MPNet (Song et al., 2020) from the Sentence Transformers library. After receiving our encoded corpus $E_\Theta(T)$, for the *Mean* and *Concat*, we construct a rule embedding r^i_Θ for each rule r_i in the ruleset. In the *Mean* strategy, this is obtained by taking the mean of all rule exemplars $\mu(r^i_\Theta) = (\frac{1}{m}\sum_i^m e^i_m)$. For *Concat*, this is calculated by concatenating all rule exemplars $\mu(r_i) = E_\Theta(e^i_1 \parallel ... \parallel e^i_m)$ and encoding the concatenated representation. Once r^i_Θ is constructed, we then label each text in the corpus whose cosine similarity is within the threshold k:

Content Moderation Using Rules (Fully Supervised)												
	HateXplain				Jigsaw				CAD			
Model	Precision	Recall	F1	Acc	Precision	Recall	F1	Acc	Precision	Recall	F1	Acc
HateXplain Rules	0.609	0.983	0.752	0.615	-	-	-	-	-	-	-	-
Hate+Abuse Rules	0.755	0.687	0.719	0.682	0.164	0.361	0.226	0.972	0.586	0.193	0.290	0.909
CAD Rules	-	-	-	-	-	-	-	-	0.110	0.842	0.194	0.325
BERT^{+}	0.808	0.841	0.824	0.787	0.459	0.729	0.563	0.987	0.445	0.421	0.433	0.893
MPNet$^{\wedge}$	0.795	0.854	0.823	0.783	0.510	0.674	0.581	0.989	0.519	0.417	0.463	0.906
Rule By Example$^{+\triangle}$	0.758	0.903	0.824	0.771	0.581	0.625	0.602	0.991	0.416	0.478	0.445	0.885
Rule By Example$^{\wedge\triangle}$	0.790	0.891	**0.837**	0.795	0.508	0.746	**0.604**	0.989	0.484	0.468	0.476	0.900
Rule By Example^{+*}	0.738	0.912	0.816	0.756	-	-	-	-	-	-	-	-
Rule By Example$^{\wedge*}$	0.779	0.893	0.832	0.786	-	-	-	-	-	-	-	-
Rule By Example$^{+\ddagger}$	-	-	-	-	-	-	-	-	0.512	0.378	0.435	0.905
Rule By Example$^{\wedge\ddagger}$	-	-	-	-	-	-	-	-	0.508	0.448	**0.476**	0.905

Table 1: Experiment Results in Fully Supervised Setting on hate speech classification datasets. $^{+}$Uses BERT (Devlin et al., 2018) as the base model. $^{\wedge}$Uses MPNet (Song et al., 2020) as the base model. *Uses HateXplain ruleset. $^{\triangle}$Uses Hate+Abuse ruleset. ‡Uses CAD Ruleset. **Note:** The HateXplain Ruleset and Contextual Abuse Dataset (CAD) Ruleset are only applicable to their respective datasets.

$$f(t_i) = \begin{cases} 1, & \text{if } sim(r^i_\Theta, E_\Theta(t_i)) \geqslant k \\ 0, & \text{otherwise} \end{cases} \quad (3)$$

In contrast to the *Mean* and *Concat* strategies, the *Distance* strategy takes a rule elimination approach. Given an unlabeled corpus $T = \{t_1, t_2, ..., t_n\}$, ruleset $R = \{(r_1, e_1), ..., (r_n, e_n)\}$, and a threshold k, we first noisily label the entire corpus using the ruleset $R_i : x_t \rightarrow \{1, \varnothing\}$ such that each rule is paired with a cover set $R = \{(r_1, e_1, c_1), ..., (r_n, e_n, c_n)\}$ where c_i is the set of texts in covered by r_i. Next, for each rule, we encode text in its cover set $E_\Theta(c_i)$ and calculate the average cosine distance between each embedding and its neighboring examples in c_i.

$$avgDist(E_\Theta(c_i)) = \frac{1}{n}\sum_{i}^{n} dist(c^i_j, c^i_{j-1}) \quad (4)$$

Lastly, once the average distance for each rule is calculated, using the defined threshold k, we flip any weakly labeled examples in the cover set if the average distance for that rule is above the threshold k:

$$f(t_i) = \begin{cases} 1, & \text{if } avgDist(r_i) \geqslant k \\ 0, & \text{otherwise} \end{cases} \quad (5)$$

4 Results and Discussion

We analyze the results of our experiments, detail our insights, and discuss the implications of applying RBE for explainable hate speech detection.

Evaluation Metrics: The precision, recall, and F1 score for each dataset in a supervised setting are reported in Table 1. Due to the highly skewed class distribution, we favor macro F1 scores as our main evaluation metric. We also report accuracy scores (the fraction of entries for which the full set of labels matches) as another metric.

4.1 Supervised Performance

Table 1 reports our results on three hate speech classification datasets in the supervised setting. We observe that RBE is able to outperform SOTA transformer-based models BERT and MPNet by 1.3/1.4%, 4.1/2.3%, and 4.3/1.3% in F1-score on HateXplain, Jigsaw, and CAD respectively. This improvement highlights the impact of leveraging rules in the training process of our framework. Additionally, it is important to note that this increase was achieved using only 1 exemplar per rule in the ruleset. These exemplars were also used to train the comparative baseline models, ensuring that all approaches were trained on the same number of samples. This further showcases how lightweight and flexible RBE is to integrate into a content moderation workflow. For HateXplain, our experiments show that the combination of MPNet as the initialized encoder with both the HateXplain Rationale and Hate+Abuse Ruleset delivers the best performance. Upon deeper analysis, we find that this is due to two main factors:

1) **Ruleset Size and Alignment** - As explained in Section 3.2 the HateXplain Rationale Ruleset was automatically crafted using rationale labels from expert annotators. This results in a powerful ruleset capable of identifying a large amount of hateful content in the HateXplain dataset as shown

Rule By Example Rule Grounding			
Dataset	**Text**	**Fired Rules**	**Exemplar**
HateXplain	fully agree every personal interaction with these queers reinforces what i already knew severe mental illness and obnoxious to boot	if contains("queers") $\rightarrow 1, 0$	yes but queers are too self righteous to let them be included originally the gay and pedophile communities were working together nambla was started by gay men who liked boys now all subsets of pedophiles are members assuming nambla still exists
Jigsaw	Why do they put so many gay people on the damn show since when it was okay to be gay.	if contains("so" $\wedge$"gay") $\rightarrow 1, 0$	stop reverting my edit your so f**** gay get a f**** life your f**** or go get laid or something
CAD	What a little b****	if contains("b****") $\rightarrow 1, 0$	Nope, today is tuna b****

Table 2: Example of Rule-grounding explanations. This table shows examples of traced model predictions produced by RBE. By displaying the rules and exemplars responsible, rule authors and users are better able to understand model predictions and can automatically adjust their ruleset to further improve model performance.

by the high recall score of the HateXplain Rationale Ruleset in Table 1. Additionally, when applied to the HateXplain dataset, the HateXplain Rationale Ruleset produces a total of 577 rules compared to the 377 rules derived from the Hate+Abuse Ruleset, allowing for more rule representations for the model to contrast against.

2) **Embedding Initialization** - Out of the box, pre-trained BERT does not produce meaningfully distinct sentence representations. In practice, the BERT [CLS] token as well as averaged BERT outputs can contain useful information after downstream fine-tuning. This is shown by the BERT performance in Table 1. However, when the pre-trained model output is pooled across all dimensions and used for calculating semantic similarity, this results in similar representations even for completely different input text. As a result, if applied to the HateXplain dataset without any fine-tuning, BERT embeddings obtain a precision, recall, and F1-score of 59%, 100%, and 75% respectively, where every example is labeled as hateful. This lack of varied sentence representation coupled with a verbose ruleset such as the HateXplain Rationale Ruleset results in an initial biasing towards hateful examples as shown by the high recall scores. As such, utilizing a pre-trained sentence embedder, such as MPNet, with a pre-train task more optimized for semantic embeddings results in better performance. We observe a similar trend when utilizing our derived ruleset for CAD. **Note:** When trained longer, the bias of the BERT model decreases as more varied sentence representations are learned.

On Jigsaw and Contextual Abuse datasets using the Hate+Abuse List and derived CAD Ruleset, RBE outperforms SOTA by an increased margin of 4.1/2.3%, and 4.3/1.3% respectively. Contrary to HateXplain, these two datasets are more heavily imbalanced toward non-hateful examples and thus more representative of the real-world case of content moderation where most content is considered benign. This increased performance highlights the power of incorporating logical rules to assist model learning and also the ability of RBE to better generalize rules. As seen in Table 1, on its own the Hate+Abuse ruleset performs poorly on each dataset in both precision and recall. Despite RBE's reliance on this ruleset to guide model learning, when combined with labeled training data, RBE is capable of both restricting over-generalized rules and leveraging its understanding of semantic similarity to extend fragile rules regardless of the base model. Additionally, when using the CAD ruleset which is heavily overfitted to the CAD dataset, as shown by the skewed recall score, RBE is still capable of outperforming the baselines.

Out-of-domain Rulesets Our Hate+Abuse ruleset is a generic ruleset unrelated to any of the datasets evaluated, and thereby an out-of-domain ruleset. This provides an example of out-of-domain performance using rules not derived from the target dataset. We observe that even when applying RBE with the Hate+Abuse ruleset we are able to outperform the baselines on each dataset. When applying RBE to new domain settings, all that is required is the authoring of additional rules for this new domain. This can be done manually, or more scalably by automatically deriving rules from the new domain data.

4.2 Interpretability

In addition to its improved performance, another advantage of RBE lies in its ability to perform Rule-grounding. As explained in section 2.2, Rule-grounding enables us to trace our model predictions back to their respective rule accompanied by the exemplars that define that rule. Table 2 shows Rule-grounding examples extracted from each of our tested datasets. By nature, Rule-grounding enables two main features in RBE:

1) **Customizability/Ruleset Adaptation**: Given the vast reach of online applications, content mod-

Content Moderation Using Rules (Unsupervised)												
	HateXplain				Jigsaw				CAD			
Model	Precision	Recall	F1	Acc	Precision	Recall	F1	Acc	Precision	Recall	F1	Acc
HateXplain Rules	0.609	0.983	0.752	0.615	-	-	-		-	-	-	
Hate+Abuse Rules	0.755	0.687	0.719	0.682	0.164	0.361	0.226	0.972	0.586	0.193	0.290	0.909
CAD Rules	-	-	-	-	-	-	-	-	0.110	0.842	0.194	0.325
BERT^{+*}	0.606	0.990	0.752	0.613	-	-	-	-	-	-	-	-
BERT$^{+\triangle}$	0.747	0.717	0.732	0.688	0.234	0.461	0.310	0.977	0.587	0.205	0.303	0.909
BERT$^{+\ddagger}$	-	-	-	-	-	-	-	-	0.107	0.865	0.191	0.290
MPNet^{+*}	0.611	0.991	0.756	0.621	-	-	-	-	-	-	-	-
MPNet$^{+\triangle}$	0.652	0.850	0.738	0.641	0.247	0.501	**0.331**	0.977	0.642	0.199	0.304	0.912
MPNet$^{+\ddagger}$	-	-	-	-	-	-	-	-	0.111	0.840	0.196	0.335
Rule By Example (Distance)*	0.614	0.983	0.756	0.623	-	-	-	-	-	-	-	-
Rule By Example (Distance)$^{\triangle}$	0.629	0.955	0.758	0.639	0.358	0.284	0.317	0.986	0.280	0.322	0.299	0.854
Rule By Example (Distance)‡	-	-	-	-	-	-	-	-	0.166	0.522	0.252	0.701
Rule By Example (Concat)*	0.621	0.950	0.751	0.626	-	-	-	-	-	-	-	-
Rule By Example (Concat)$^{\triangle}$	0.612	0.985	0.755	0.621	0.189	0.052	0.081	0.987	0.175	0.437	0.250	0.747
Rule By Example (Concat)‡	-	-	-	-	-	-	-	-	0.178	0.437	0.253	0.750
Rule By Example (Mean)*	0.612	0.983	0.754	0.620	-	-	-	-	-	-	-	-
Rule By Example (Mean)$^{\triangle}$	0.636	0.944	0.760	0.646	0.188	0.124	0.149	0.984	0.294	0.273	0.283	0.866
Rule By Example (Mean)‡	-	-	-	-	-	-	-	-	0.189	0.411	0.259	0.772
Unsupervised Pre-Training												
Rule By Example (Mean)$^{\triangle}$	0.641	0.954	**0.767**	0.656	0.166	.626	0.262	0.961	0.260	0.320	0.287	0.846
Rule By Example (Distance)$^{\triangle}$	0.617	0.968	0.753	0.624	0.203	0.465	0.283	0.974	0.484	0.236	**0.317**	0.902

Table 3: Unsupervised Performance across all clustering strategies. *Uses HateXplain ruleset. $^{\triangle}$Uses Hate+Abuse ruleset. ‡Uses CAD ruleset. **Note:** The HateXplain Ruleset is not applicable to Jigsaw and Contextual Abuse Dataset (CAD).

eration systems need to be easily adaptable to ever-emerging trends of hateful content. Particularly in online social settings, expert users of these platforms continually find new and interesting ways to bypass moderation systems. Additionally, new terminologies and slang are being introduced every day. RBE is seamlessly capable of addressing these concerns by facilitating rule-guided learning. By defining a new rule and adding at least one exemplar, RBE is able to capture emerging content without the need for re-training. Additionally, users of RBE can easily modify existing rules that may be too broad and add additional exemplars to further refine predictions in a controllable manner.

2) **Prediction Transparency**: By facilitating model interpretations via rule-grounding, users of online systems are offered tangible guidance should their content be flagged, potentially increasing user trust in the system. Additionally, this acts as a direct indicator of the type of content the rule authors want to moderate.

4.3 Unsupervised Performance

Table 3 reports our results in the unsupervised setting. We observe that RBE is able to outperform SOTA trained on noisy rules labeled samples for the HateXplain and Jigsaw dataset while also outperforming the ruleset as is on all three datasets. Across each dataset, we find that RBE's *Distance* based strategy produces the most consistent performance, outperforming SOTA on HateXplain and CAD while performing on par with SOTA on Jigsaw. We observe that this stability in performance is due to this strategy's rule elimination objective. As opposed to the *Mean* and *Concat* strategies which focus on deriving rule representations in a self-supervised manner, the *Distance* strategy instead focuses on eliminating over-generalized rules whose cover set of examples are semantically dissimilar. This is particularly useful in cases where precision scores are low due to a large number of false positives.

For Jigsaw, we observe a slight decrease in performance compared to SOTA. Upon further analysis, we posit that this is a result of RBE's over-reliance on the ruleset in this setting, particularly for the *Mean* and *Concat* strategies. This is because the ruleset directly influences the derived rule embedding due to its labeling of the cover set C. As such when the ruleset is over-generalized, as is the case of Hate+Abuse rules on Jigsaw, RBE is likely to match the distribution of the ruleset. We find that performing self-supervised model pre-training (Gao et al., 2021) on the target corpus circumvents this trend for the *Mean* and *Concat* strategy. As such, with a more refined ruleset, a performance increase is expected as seen in HateXplain and CAD.

5 Related Work

There has been active work on detecting hate speech in language (Poletto et al., 2021; Al-Makhadmeh and Tolba, 2020; Schmidt and Wie-

gand, 2017). Hate Speech detection has proven to be a nuanced and difficult task, leading to the development of approaches and datasets targeted at various aspects of the problem (Vidgen et al., 2021; Mathew et al., 2020; Mody et al., 2023). However, few attempts have been made to focus on the explainability of these models, which is an increasing area of concern surrounding their use online (Tarasov, 2021; Haimson et al., 2021), thus leading to the continued utilization of less powerful but more explainable methods such as rules. Prior works have explored incorporating logical rules into model learning. Awasthi et al. (2020) proposed to weakly learn from rules by pairing them with exemplars and training a denoising model. However, this requires defining rules for all output classes, making it inapplicable to the task of hate speech detection. Additionally, this method only focuses on decreasing rule scope to solve the over-generalization problem. It does not simultaneously tackle the over-specificity problem demonstrated in Figure 1. Finally, this method does not provide a way for interpreting model predictions during inference. Seo et al. (2021) proposes a way to control neural network training and inference via rules, however, their framework represents rules as differentiable functions requiring complex perturbations to incorporate, making it more suitable to numerical rules such as those defined in healthcare and finance as opposed to the complex nuances of language. Pryzant et al. (2022) proposes a framework for the automatic induction of symbolic rules from a small set of labeled data. However, these rules are derived from low-capacity ML models and are as a result not human-readable or explainable.

6 Conclusion

We introduce Rule By Example, an exemplar-based contrastive learning framework that enables learning from logical rules for accurate and explainable hate speech detection. Specifically, we propose a novel dual-encoder model architecture designed to produce meaningful rule and text representations. RBE leverages a novel exemplar-based contrastive learning objective that converges the representations of rules and text inputs of similar classes. We share results on three public datasets for hate speech detection that validate the Rule By Example framework can not only vastly outperform the initial ruleset but also outperform baseline SOTA classification methods in both supervised and unsupervised settings. Moreover, RBE enables rule-grounding which allows for more explainable model prediction benefits not available in SOTA classification methods alongside additional flexibility via Ruleset Adaptation.

7 Limitations

In this section, we discuss some of the limitations of the Rule by Example method.

7.1 Dependence on Supervision

The requirement of both a set of rules and an example per rule in our Rule by Example method means that some amount of expert supervision is required, even for the 'unsupervised' experimental setups. This could be a prohibitive cost in some scenarios. There are potential methods to select an example per rule in an unsupervised manner, such as clustering the examples the rules fires on, that could be explored in future work. However, the creation of the rules themselves means some form of expert supervision that distills knowledge about the classification task into a parseable function.

7.2 Increased Cost Compared to Rules

Although the Rule by Example method produces a Dual Encoder model that is shown to be much more performant than the ruleset it is derived from, it still has the cost limitations of other deep learning methods. The Dual Encoder requires far more expensive compute (GPUs) to initially train and later inference in a production setting. And even with using expensive GPUs, the latency cost is unavoidably much higher than most simple logical rules. For some applications, the quality gain of the Dual Encoder model may not be worth the increased operational cost.

7.3 Reliance on Quality Rules and Exemplars

Since the Rule by Example method is based on having a ruleset and associated exemplars to learn from, the quality of those rules and exemplars could affect downstream Dual Encoder model quality. If the authored ruleset and chosen exemplars are not high quality, intuitively the quality of the Dual Encoder model would suffer. This is especially true in the unsupervised setting, where the rules are used as noisy labeling functions. A possible future extension is studying the effect of rule and exemplar quality on the performance of the derived Dual Encoder model.

8 Ethics

Hate speech detection is a complex task. Reducing the task to authoring a set of simple logical rules can potentially lead to rule authors encoding hard biases in those rules. This can cause problems of erasure, for example, if an in-group word or an identity term is used as a rule to identify content as hate speech.

The Rule by Example method can potentially reduce these cases, for example by learning a better rule representation and identifying when a term is used as in-group speech as opposed to being used as an insult or slur. However, the derived Dual Encoder is also at the risk of propagating and amplifying these biases (Hall et al., 2022), causing greater unintended harm than the original ruleset.

Whether using a ruleset or using a more complicated model, it is important to support classifiers with additional Responsible AI work streams, such as reviews of classifier behavior and measurements of fairness.

Acknowledgements

We thank our anonymous reviewers for their feedback and suggestions. This work was conducted by the ROAR (Responsible & Open AI Research) team at Microsoft Cloud & AI. At UofM, Christopher Clarke is supported in part by award NSF1539011 by the National Science Foundation.

References

Zafer Al-Makhadmeh and Amr Tolba. 2020. Automatic hate speech detection using killer natural language processing optimizing ensemble deep learning approach. *Computing*, 102(2):501–522.

Abhijeet Awasthi, Sabyasachi Ghosh, Rasna Goyal, and Sunita Sarawagi. 2020. Learning from rules generalizing labeled exemplars.

Christopher Clarke, Joseph Peper, Karthik Krishnamurthy, Walter Talamonti, Kevin Leach, Walter Lasecki, Yiping Kang, Lingjia Tang, and Jason Mars. 2022. One agent to rule them all: Towards multi-agent conversational AI. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 3258–3267, Dublin, Ireland. Association for Computational Linguistics.

Emmanuel Gbenga Dada, Joseph Stephen Bassi, Haruna Chiroma, Shafi'i Muhammad Abdulhamid, Adebayo Olusola Adetunmbi, and Opeyemi Emmanuel Ajibuwa. 2019. Machine learning for email spam filtering: review, approaches and open research problems. *Heliyon*, 5(6):e01802.

Thomas Davidson, Dana Warmsley, Michael Macy, and Ingmar Weber. 2017. Automated hate speech detection and the problem of offensive language.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. Simcse: Simple contrastive learning of sentence embeddings.

Tarleton Gillespie. 2018. *Custodians of the Internet: Platforms, content moderation, and the hidden decisions that shape social media*. Yale University Press.

R. Hadsell, S. Chopra, and Y. LeCun. 2006. Dimensionality reduction by learning an invariant mapping. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 2, pages 1735–1742.

Oliver L. Haimson, Daniel Delmonaco, Peipei Nie, and Andrea Wegner. 2021. Disproportionate removals and differing content moderation experiences for conservative, transgender, and black social media users: Marginalization and moderation gray areas. *Proc. ACM Hum.-Comput. Interact.*, 5(CSCW2).

Melissa Hall, Laurens van der Maaten, Laura Gustafson, Maxwell Jones, and Aaron Adcock. 2022. A systematic study of bias amplification.

Prerna Juneja, Deepika Rama Subramanian, and Tanushree Mitra. 2020. Through the looking glass: Study of transparency in reddit's moderation practices. *Proc. ACM Hum.-Comput. Interact.*, 4(GROUP).

David Kemp and Emily Ekins. 2021. Poll: 75% don't trust social media to make fair content moderation decisions, 60% want more control over posts they see.

Vivian Lai, Samuel Carton, Rajat Bhatnagar, Q. Vera Liao, Yunfeng Zhang, and Chenhao Tan. 2022. Human-ai collaboration via conditional delegation: A case study of content moderation. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*, CHI '22, New York, NY, USA. Association for Computing Machinery.

Kevin Lee. 2022. Rules vs. machine learning: Why you need both to win: Sift.

Yi Liu, Pinar Yildirim, and Z. John Zhang. 2022. Implications of revenue models and technology for content moderation strategies. *Marketing Science*, 41(4):831–847.

Jitendra Singh Malik, Guansong Pang, and Anton van den Hengel. 2022. Deep learning for hate speech detection: A comparative study.

Binny Mathew, Punyajoy Saha, Seid Muhie Yimam, Chris Biemann, Pawan Goyal, and Animesh Mukherjee. 2020. Hatexplain: A benchmark dataset for explainable hate speech detection.

Devansh Mody, YiDong Huang, and Thiago Eustaquio Alves de Oliveira. 2023. A curated dataset for hate speech detection on social media text. *Data in Brief*, 46:108832.

Animesh Mukherjee, Mithun Das, Binny Mathew, and Punyajoy Saha. 2022. Hate speech: Detection, mitigation and beyond @aaai.

Fabio Poletto, Valerio Basile, Manuela Sanguinetti, Cristina Bosco, and Viviana Patti. 2021. Resources and benchmark corpora for hate speech detection: a systematic review. *Language Resources and Evaluation*, 55(2):477–523.

Reid Pryzant, Ziyi Yang, Yichong Xu, Chenguang Zhu, and Michael Zeng. 2022. Automatic rule induction for interpretable semi-supervised learning.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks.

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*, pages 1–10, Valencia, Spain. Association for Computational Linguistics.

Sungyong Seo, Sercan O. Arik, Jinsung Yoon, Xiang Zhang, Kihyuk Sohn, and Tomas Pfister. 2021. Controlling neural networks with rule representations.

Yusuke Shido, Hsien-Chi Liu, and Keisuke Umezawa. 2022. Textual content moderation in C2C marketplace. In *Proceedings of the Fifth Workshop on e-Commerce and NLP (ECNLP 5)*, pages 58–62, Dublin, Ireland. Association for Computational Linguistics.

Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. 2020. Mpnet: Masked and permuted pre-training for language understanding.

Katie Tarasov. 2021. Why content moderation costs billions and is so tricky for facebook, twitter, youtube and others.

Sahaj Vaidya, Jie Cai, Soumyadeep Basu, Azadeh Naderi, Donghee Yvette Wohn, and Aritra Dasgupta. 2021. Conceptualizing visual analytic interventions for content moderation. In *2021 IEEE Visualization Conference (VIS)*, pages 191–195.

Bertie Vidgen, Dong Nguyen, Helen Margetts, Patricia Rossini, and Rebekah Tromble. 2021. Introducing CAD: the contextual abuse dataset. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2289–2303, Online. Association for Computational Linguistics.

Kurt Wagner and Bloomberg. 2021. Facebook says it has spent $13 billion on safety and security efforts since 2016.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Canwen Xu, Daya Guo, Nan Duan, and Julian McAuley. 2022. Laprador: Unsupervised pretrained dense retriever for zero-shot text retrieval.

Yuchen Zhang. 2019. Stop bad content before it's posted, and build better communities.

ACL 2023 Responsible NLP Checklist

A For every submission:

☑ A1. Did you describe the limitations of your work?
7

☑ A2. Did you discuss any potential risks of your work?
8

☑ A3. Do the abstract and introduction summarize the paper's main claims?
1

☒ A4. Have you used AI writing assistants when working on this paper?
Left blank.

B ☑ Did you use or create scientific artifacts?

2

☐ B1. Did you cite the creators of artifacts you used?
No response.

☐ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
No response.

☐ B3. Did you discuss if your use of existing artifact(s) was consistent with their intended use, provided that it was specified? For the artifacts you create, do you specify intended use and whether that is compatible with the original access conditions (in particular, derivatives of data accessed for research purposes should not be used outside of research contexts)?
No response.

☐ B4. Did you discuss the steps taken to check whether the data that was collected / used contains any information that names or uniquely identifies individual people or offensive content, and the steps taken to protect / anonymize it?
No response.

☐ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
No response.

☑ B6. Did you report relevant statistics like the number of examples, details of train / test / dev splits, etc. for the data that you used / created? Even for commonly-used benchmark datasets, include the number of examples in train / validation / test splits, as these provide necessary context for a reader to understand experimental results. For example, small differences in accuracy on large test sets may be significant, while on small test sets they may not be.
Left blank.

C ☑ Did you run computational experiments?

3

☑ C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
3

The Responsible NLP Checklist used at ACL 2023 is adopted from NAACL 2022, with the addition of a question on AI writing assistance.

☑ C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?
3

☑ C3. Did you report descriptive statistics about your results (e.g., error bars around results, summary statistics from sets of experiments), and is it transparent whether you are reporting the max, mean, etc. or just a single run?
4

☑ C4. If you used existing packages (e.g., for preprocessing, for normalization, or for evaluation), did you report the implementation, model, and parameter settings used (e.g., NLTK, Spacy, ROUGE, etc.)?
3

D ☒ Did you use human annotators (e.g., crowdworkers) or research with human participants?

Left blank.

☐ D1. Did you report the full text of instructions given to participants, including e.g., screenshots, disclaimers of any risks to participants or annotators, etc.?
No response.

☐ D2. Did you report information about how you recruited (e.g., crowdsourcing platform, students) and paid participants, and discuss if such payment is adequate given the participants' demographic (e.g., country of residence)?
No response.

☐ D3. Did you discuss whether and how consent was obtained from people whose data you're using/curating? For example, if you collected data via crowdsourcing, did your instructions to crowdworkers explain how the data would be used?
No response.

☐ D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?
No response.

☐ D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?
No response.