Explainable Abuse Detection as Intent Classification and Slot Filling

Agostina Calabrese and Björn Ross and Mirella Lapata
Institute for Language, Cognition and Computation
School of Informatics, University of Edinburgh
10 Crichton Street, Edinburgh EH8 9AB, United Kingdom
{a.calabrese@,b.ross@,mlap@inf.}ed.ac.uk

Abstract

To proactively offer social media users a safe online experience, there is a need for systems that can detect harmful posts and promptly alert platform moderators. In order to guarantee the enforcement of a consistent policy, moderators are provided with detailed guidelines. In contrast, most state-of-the-art models learn what *abuse* is from labeled examples and as a result base their predictions on spurious cues, such as the presence of group identifiers, which can be unreliable. In this work we introduce the concept of policy-aware abuse detection, abandoning the unrealistic expectation that systems can reliably learn which phenomena constitute abuse from inspecting the data alone. We propose a machine-friendly representation of the policy that moderators wish to enforce, by breaking it down into a collection of intents and slots. We collect and annotate a dataset of 3,535 English posts with such slots, and show how architectures for intent classification and slot filling can be used for abuse detection, while providing a rationale for model decisions.[1]

1 Introduction

The central goal of online content moderation is to offer users a safer experience by taking actions against abusive behaviors, such as hate speech. Researchers have been developing supervised classifiers to detect hateful content, starting from a collection of posts known to be abusive and non-abusive. To successfully accomplish this task, models are expected to learn complex concepts from previously flagged examples. For example, hate speech has been defined as ''abusive speech targeting specific group characteristics, such as ethnic origin, religion, gender or sexual orientation'' (Warner and Hirschberg, 2012), but there is no clear definition of what constitutes *abusive speech*.

Recent research (Dixon et al., 2018) has shown that supervised models fail to grasp these complexities; instead, they exploit spurious correlations in the data, they become overly reliant on low-level lexical features and flag posts because of, for instance, the presence of group identifiers alone (e.g., *women* or *gay*). Efforts to mitigate these problems focus on regularization, for example, preventing the model from paying attention to group identifiers during training (Kennedy et al., 2020; Zhang et al., 2020), however, they do not seem effective at producing better classifiers (Calabrese et al., 2021). Social media companies, on the other hand, give moderators detailed guidelines to help them decide whether a post should be deleted, and these guidelines also help ensure consistency in their decisions (see Table 1). Models are not given access to these guidelines, and arguably this is the reason for many of their documented weaknesses.

Let us illustrate this with the following example. Assume we are shown two posts, the abusive ''*Immigrants are parasites*'', and the non-abusive ''*I love artists*'', and are asked to judge whether a new post ''*Artists are parasites*'' is abusive. Although the post is insulting, it does not contain hate speech, as professions are not usually protected, but we cannot know that without access to moderation guidelines. Based on these two posts alone, we might struggle to decide which label to assign. We are then given more examples, specifically the non-abusive ''*I hate artists*'' and the abusive ''*I hate immigrants*''. In the absence of any other information, we would probably label the post ''*Artists are parasites*'' as non-abusive. The example highlights that 1) the current problem formulation (i.e., given post p and a collection of labeled examples C, decide whether p

[1]Our code and data are available at https://github.com/Ago3/PLEAD.

Transactions of the Association for Computational Linguistics, vol. 10, pp. 1440–1454, 2022. https://doi.org/10.1162/tacl_a_00527
Action Editor: Nitin Madnani. Submission batch: 5/2022; Revision batch: 8/2022; Published 12/2022.

Post: Artists are parasites
Policy: Posts containing dehumanizing comparisons targeted to a group based on their protected characteristics violate the policy. Protected characteristics include race, ethnicity, national origin, disability, religious affiliation, caste, sexual orientation, sex, gender identity, serious disease, and immigration status.
Old Formulation: Is the post abusive? **Our Formulation:** Does the post violate the policy?

Table 1: Although it is hard to judge whether a post is abusive based solely on its content, taking the policy into account facilitates decision making. The example is based on the Facebook Community Standards.

is abusive) is not adequate, since even humans would struggle to agree on the correct classification, and 2) relying on group identifiers is a natural consequence of the problem definition, and often not incorrect. Note that the difficulty does not arise due to the lack of data annotated with real moderator decisions who would be presumably making labeling decisions according the policy. Rather, models are not able to distinguish between necessary and sufficient conditions for making a decision based on examples alone (Balkir et al., 2022).

In this work we depart from the common approach that aims to mitigate undesired model behavior by adding artificial constraints (e.g., ignoring group identifiers when judging hate speech) and instead re-define the task through the concept of *policy-awareness*: given post p *and* policy P, decide whether p violates P. This entails that models are given policy-related information in order to classify posts like *''Artists are parasites''*; for example, they know that posts containing dehumanizing comparisons targeted to a group based on their protected characteristics violate the policy, and that profession is not listed among the protected characteristics (see Table 1). To enable models to exploit the policy, we formalize the task as an instance of intent classification and slot filling and create a *machine-friendly representation* of a policy for hate speech by decomposing it into a collection of intents and corresponding slots. For instance, the policy in Table 1 expresses the intent ''Dehumanization'' and has three slots: ''target'', ''protected characteristic'', and ''dehumanizing comparison''. All slots must be present for a post to violate a policy. Given this definition, the post in Table 1 contains a target (*''Artists''*) and a dehumanizing comparison (*''are parasites''*) but does not violate the policy since it does not have a value for protected characteristic.

We create and make publicly available the **Po**licy-aware **E**xplainable **A**buse **D**etection (PLEAD) dataset, which contains (intent and slot) annotations for 3,535 abusive and non-abusive posts. To decide whether a post violates the policy and explain the decision, we design a sequence-to-sequence model that generates a structured representation of the input by first detecting and then filling slots. Intent is assigned deterministically based on the filled slots, leading to the final abusive/non-abusive classification. Experiments show our model is more reliable than classification-only approaches, as it delivers transparent predictions.

2 Related Work

We use *abuse* as an umbrella term covering any kind of harmful content on the Web, as this is accepted practice in the field (Vidgen et al., 2019; Waseem et al., 2017). Abuse is hard to recognize, due to ambiguity in its definition and differences in annotator sensitivity (Ross et al., 2016). Recent research suggests embracing disagreements by developing multi-annotator architectures that capture differences in annotator perspective (Davani et al., 2022; Basile et al., 2021; Uma et al., 2021). While this approach better models how abuse is perceived, it is not suitable for content moderation where one has to decide whether to remove a post and a prescriptive paradigm is preferable (Röttger et al., 2022).

Zufall et al. (2020) adopt a more objective approach, as they aim to detect content that is *illegal* according to EU legislation. However, as they explain, illegal content constitutes only a tiny portion of abusive content, and no explicit knowledge about the legal framework is provided to their model. The problem is framed as the combination of two binary tasks: whether a post contains a protected characteristic, and whether it incites violence. The authors also create a dataset which, however, is not publicly available.

Most existing work ignores these annotation difficulties and models abuse detection with transformer-based models (Vidgen et al., 2021b;

Intent	Policy Definition	Examples and Slot Annotations
Dehumanization	Posts containing dehumanizing comparisons targeted to an individual or group based on their protected characteristics violate the policy.	Immigrants are parasites [Target *Immigrants*], [ProtectedCharacteristic *Immigrants*], [DehumanisingComparison *are parasites*]
Threatening	Posts containing threatening language targeted to an individual or group based on their protected characteristics violate the policy.	"Women should be shot? Agreed!"* [Target *Women*], [ProtectedCharacteristic *Women*], [ThreateningSpeech *should be shot*]
Derogation	Posts containing derogatory terms or insults targeted to an individual or group based on their protected characteristics violate the policy.	"Truly, I have been to Poland and there's nothing to see there. Polish people are so nasty as well." [Target *Polish people*], [ProtectedCharacteristic *Polish*], [DerogatoryOpinion *are so nasty*]
Animosity	Posts containing *implicit* abusive language targeted to an individual or group based on their protected characteristics violate the policy.	"I don't want to have to be around a breastfeeding woman, if she wants to do that she can do it at home" [Target *a breastfeeding woman*], [ProtectedCharacteristic *she*], [NegativeOpinion *can do it at home*]
Pro Hate Crime	Posts that glorify, support or deny hateful actions, events, organisations and individuals.	"Hitler was right all along. We are witnessing it at home EVERY day." [HateEntity *Hitler*], [Support *was right all along*]

Table 2: Definition of policy guidelines, intents, and slots associated with them. Example posts and their annotations. Wording in the guidelines that is mapped onto slots is underlined.

Kennedy et al., 2020; Mozafari et al., 2019). Despite impressive F1-scores, these models are black-box and not very informative for moderators. Efforts to shed light on their behavior, reveal that they are good at exploiting spurious correlations in the data but unreliable in more realistic scenarios (Calabrese et al., 2021; Röttger et al., 2021). Although explainability is considered a critical capability (Mishra et al., 2019) in the context of abuse detection, to our knowledge, Sarwar et al. (2022) represent the only explainable approach. Their model justifies its predictions by returning the k nearest neighbors that determined the classification outcome. However, such ''explanations'' may not be easily understandable to humans, who are less skilled at detecting patterns than transformers (Vaswani et al., 2017).

In our work, we formalize the problem of policy-aware abuse detection as an instance of intent classification and slot filling (ICSF), where slots are properties like ''target'' and ''protected characteristic'' and intents are policy rules or guidelines (e.g., ''dehumanization''). While Ahmad et al. (2021) use ICSF to parse and explain the content of a privacy policy, we are not aware of any work that infers policy violations in text with ICSF. State-of-the-art models developed for ICSF are sequence-to-sequence transformers built on top of pretrained architectures like BART (Aghajanyan et al., 2020), and also represent the starting point for our modeling approach.

3 Problem Formulation

Given a policy for the moderation of abusive content, and a post p, our task is to decide whether p is abusive. We further note that policies are often expressed as a set of guidelines $R = \{r_1, r_2, \ldots r_N\}$ as shown in Table 2 and a post p is abusive when its content violates any $r_i \in R$. Aside from deciding whether a guideline has been violated, we also expect our model to return a human-readable explanation that should be specific to p (i.e., an extract from the policy describing the guideline being violated is not an explanation), since customized explanations can help moderators make more informed decisions and developers better understand model behavior.

Intent Classification and Slot Filling The generation of post-specific explanations requires detection systems to be able to reason over the content of the policy. To facilitate this process, we draw inspiration from previous work (Gupta et al., 2018) on ICSF, a task where systems have to classify the intent of a query (e.g., `IN:CREATE_CALL` for the query *''Call John''*) and fill the slot associated with it (e.g., *''Call''* is the filler for the slot `SL:METHOD` and *''John''* for `SL:CONTACT`). For our task, we decompose policies into a collection of intents corresponding to the guidelines mentioned above, and each intent is characterized by a set of properties, namely, slots (see Table 2).

The canonical output of ICSF systems is a tree structure. Multiple representations have been defined, each with a different trade-off between expressivity and ease of parsing. For our use case, we adopt the decoupled representation proposed in Aghajanyan et al. (2020): Non-terminal nodes are either slots or intents, the root node is an intent, and terminal nodes are words attested in the post (see Figure 1). In this representation,

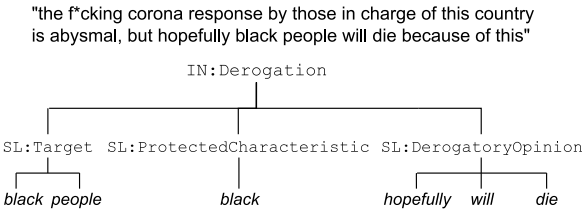

Figure 1: Decoupled representation for a post.

it is not necessary for all input words to appear in the tree (i.e., in-order traversal of the tree cannot reconstruct the original utterance). Although this ultimately renders the parsing task harder, it is crucial for our domain where words can be associated with multiple slots or no slots, and reasoning over long-term dependencies is necessary to recognize, for example, a derogatory opinion (see Figure 1).

Importantly, we first identify the slots occurring in a post and then deterministically infer the author's intent, as this renders the output tree an *explanation* of the final classification outcome rather than a post-hoc *justification* (Biran and Cotton, 2017). Likewise, since we view the predicted slots as an explanation for intent, we cannot jointly perform intent classification and slot filling, to avoid producing inconsistent explanations (Camburu et al., 2020; Ye and Durrett, 2022).

Hate Speech Taxonomy As a case-study, we model the codebook[2] for hate speech annotations designed by the Alan Turing Institute (Vidgen et al., 2021b). This policy is very similar to the guidelines that social media platforms provide to moderators and users.[3]

We obtained an intent from each section of the policy, and associated it with a set of slots (see Table 2). We followed the policy guidelines closely and slots were mostly extracted verbatim from them (see underlined policy terms in Table 2 which give rise to slots). We refrained from renaming or grouping slots to create more abstract labels (e.g., using `SL:AbusiveSpeech` to replace `SL:DehumanisingComparison`, `SL:ThreateningSpeech`, `SL:Derogatory-Opinion`, and `SL:NegativeOpinion`). Note that commonsense knowledge is required to decide whether a span is the right filler for a slot. For instance, [`SL:ThreateningSpeech` *dog*] would be odd, while [`SL:Threatening-Speech` *should be shot*] would not.

In addition to slots corresponding to different types of hate speech, most intents have a `Target` who is being abused because of a `Protected-Characteristic`. In contrast to previous work (Sap et al., 2020; Ousidhoum et al., 2019), we distinguish targets from protected groups, as this allows annotators to better infer the target's characteristics from context. A post is deemed abusive (i.e., violates the policy) if and only if all slots for at least one of the (hateful) intents are filled. We also introduce a new intent (i.e., `IN:NotHateful`) to accommodate all posts that do not violate the policy.

Besides being more machine-friendly, our formulation is advantageous in reducing the amount of abusive instances required for training, since a model can learn to predict slots even from non-abusive instances (e.g., slots `SL:Target` and `SL:DehumanisingComparison` are also present in the non-abusive *''Artists are parasites''*). This is particularly important in this domain, since in absolute terms, abusive posts are (luckily) relatively infrequent compared to non-abusive ones (Founta et al., 2018), and most harmful content is detected by moderators and subsequently deleted.

Counter Speech In a few cases, posts might quote hate speech, but the authors clearly distance themselves from the harmful message. To enable models to correctly recognize counter speech—speech that directly counters hate, for example, by presenting facts or reacting with humor (Mathew et al., 2019)—we introduce a new slot encoding the author's stance (i.e., `SL:NegativeStance`). For instance, the post *''It's nonsense to say that Polish people are nasty''* expresses a derogatory opinion that is based on a protected characteristic of a target (i.e., ''*Polish people*''). Even though all slots for the `Derogation` intent are filled, the post is not abusive as the author is reacting to the hateful message. A post is hateful if and only if there are fillers for all associated slots but not for `SL:NegativeStance`.

[2] `https://github.com/bvidgen/Dynamically-Generated-Hate-Speech-Dataset`.

[3] e.g., `https://transparency.fb.com/en-gb/policies/community-standards/hate-speech`.

	Dehumanization			Threatening			Derogation			Pro Hate Crime			Not Hateful		
	N°	LCS (%)	A (%)	N°	LCS (%)	A (%)	N°	LCS (%)	A (%)	N°	LCS (%)	A (%)	N°	LCS (%)	A (%)
Target	972	63.38	97.32	610	71.46	98.60	1102	64.91	97.91	0	—	—	836	56.75	95.16
ProtectedCharacteristic	1006	75.50	95.02	639	82.44	97.08	1156	78.63	95.16	0	—	—	139	68.45	93.60
DehumanisingComparison	883	50.99	96.32	0	—	—	0	—	—	0	—	—	36	60.42	97.22
ThreateningSpeech	0	—	—	585	56.89	97.55	0	—	—	0	—	—	48	51.45	96.63
DerogatoryOpinion	0	—	—	0	—	—	994	45.56	93.50	0	—	—	378	48.97	92.98
HateEntity	0	—	—	0	—	—	0	—	—	173	69.87	94.64	1	100.00	33.33
Support	0	—	—	0	—	—	0	—	—	173	44.44	90.71	0	—	—
NegativeStance	0	—	—	0	—	—	0	—	—	0	—	—	40	53.87	94.02
Number of instances	883			585			994			173			900		

Table 3: Number of occurrences per slot for each intent; inter-annotator agreement measured by Longest Common Subsequence score (LCS), and percentage of annotations approved by expert (A).

4 The PLEAD Dataset

Post Selection To validate our problem formulation and for model training we created a dataset consisting of posts with slot annotations (e.g., `Target`, `ThreateningSpeech`). We built our annotation effort on an existing dataset associated with the policy guidelines introduced in Section 3 and extended it with additional span-level labels. This dataset (Vidgen et al., 2021b) was created by providing annotators with a classification model trained on 11 other datasets, and asking them to write hateful and non-hateful sentences such that they fooled the model in predicting the opposite class (hateful for non-hateful and vice versa). The process was iterative, we used sentences from the second round onwards, which were annotated with policy violations.

The dataset is not balanced, that is, some policies are violated more frequently than others. To mitigate this and reduce annotation costs, we selected all posts from the less popular policies and a random sample of posts from the most popular ones. We further merged posts annotated with derogation and animosity classes as they are similar, the main difference being the extent to which the negative opinion is implied. The number of selected posts per intent is shown in Table 3. We note that this is a collection of hard examples, as they were written so as to fool a state-of-the-art model. Most non-abusive posts in the dataset have annotations for all slots save one, or they contain counter speech and are easily confusable with hate speech.

Annotation Task We performed two annotation tasks, one for hateful posts and one for non-hateful ones. For hateful posts, annotators were presented with the post, information about the target(s), its characteristics, and the slots. They were then asked to specify the spans of text corresponding to each slot. The dataset already contains annotations about which policy is being violated. For instance, for posts labeled as `Pro Hate Crime`, annotators look for spans corresponding to `HateEntity` and `Support`. Information about the target and its characteristics is also present in metadata distributed with the dataset, and we used it to steer annotators towards a correct reading of the posts. In general the original posts, metadata, and labels are of high-quality; Vidgen et al. (2021b) report extremely high agreement for instances created during round 2, moderate for the following rounds and disagreements were resolved by an expert annotator.

Each post can contain multiple targets, and each target can be associated with multiple protected characteristics (e.g., *black woman* indicates both the race and gender of a target). Our annotation scheme assumes that only one opinion is annotated for each post. For instance, the post ''*I love black people but hate women*'' contains both a non-hateful and hateful opinion, but we only elicit annotations for the hateful one. Likewise, when a post contains more than one hateful opinion,[4] annotators select the one that better fits the associated policy and target description. Equally, for non-hateful posts, we asked annotators to focus on a single opinion, with a preference for opinions that resemble hateful messages (e.g., the second opinion in ''*I love cats, but I wish all wasps dead*''). Annotators could specify as many spans (and associated slots) as they thought appropriate, including none. If enough elements were selected for a post to violate a rule (e.g., both `HateEntity` and `Support` were specified), annotators were asked whether the post contained counter speech (and if so, to

[4]Manual inspection of a sample of hateful instances revealed the percentage of instances with multiple hateful opinions to be ∼3%.

specify a span of text for `NegativeStance`) or derogatory terms used as reclaimed identity terms (e.g., the n-word used by a member of the Black community).

Annotator Selection We recruited annotators resident in English-speaking countries through the Amazon Mechanical Turk crowdsourcing platform. To ensure high-quality annotations we designed a quiz for each policy rule and assessed the fairness of the quiz through a two-phase pilot study: In the first phase annotators were shown the instructions and asked to annotate eight sentences. These annotations were then used as possible correct answers for the quiz or to clarify the instructions. During the second phase, *new* annotators were shown the updated instructions and asked to pass a quiz consisting of three questions.

The pilot showed that most crowdworkers who understood the task were able to pass the quiz, but no one was able to pass the quiz without understanding the task. Only successful annotators were granted a *guideline-specific* qualification that allowed them to annotate real instances. To enforce consistency, annotators were prompted to pass a quiz after every 30-post batch, and each batch contained posts associated with one rule. To ensure the data was annotated correctly, we included two control questions in each batch. These were not simple attention checks, but regular posts for which the correct answers were known (from the pilot study). For the annotation of non-hateful posts, we only admitted annotators who had submitted at least 300 annotations for hateful posts, and used the first batch of annotations as a further qualification test. Overall, 75% of annotations were produced by women, 91% by people who identify as straight, and 75% by people with ethnicity other than white. We release a full breakdown of demographic information with our dataset.

Annotation Quality We collected three annotations per post. We measured inter-annotator agreement, as the length of the longest common subsequence (LCS) normalized by the length of the longest span. LCS better reflects our annotation task compared to more standard measures like Cohen's Kappa (McHugh, 2012) which could be calculated on the token level. However, our annotators do not consider words indvidually, they primarily make decisions for sequences of tokens. Aside from presuming a different annotation task, token-based Kappa leads to extremely unbalanced data with the un-annotated tokens vastly outnumbering the labels of interest, artificially inflating agreement.

As shown in Table 3, average LCS scores are high for relations like `Target` and `ProtectedCharacteristic`, but drop to $\sim$45% for relations like `DerogatoryOpinion`. After a quick inspection, we discovered that poor scores were the result of non-overlapping but nevertheless valid annotations. For instance, in ''*Homosexuality means the end of the family. The next step is legalizing pedophilia.*'', both '''*means the end of the family*'' and ''*legalizing pedophilia*'' were annotated as derogatory opinions, but since there is no overlap between the spans, agreement is 0. To better assess the quality of the annotations, an expert annotator manually reviewed 50% in each category. As shown in Table 3, the percentage of valid annotations was remarkably high, which leads us to conclude that they are reliable.

A Note on Ethics To protect annotators from exposure to hateful content, we tagged our project with the ''offensive content'' label on Amazon Mechanical Turk, included a warning in the task title, and asked for consent twice (first at the end of the information sheet, and then with a one-sentence checkbox). Annotators were presented with small batches of 30 sentences, and invited to take a break at the end of each session. They were also offered the option to quit anytime during the session, or to abandon the study at any point. A reminder to seek help in case they experienced distress was provided at the beginning of each session. The study was approved by the relevant ethics committee (details removed for anonymous peer review).

5 Abuse Detection Model

ICSF is traditionally modeled as a sequence-to-sequence problem where the input utterance represents the *source* sequence, and the *target* sequence is a linearized version of the corresponding tree. For instance, the linearized version of the tree in Figure 1 would be: [`IN:Derogation`, [`SL:Target`, *black, people*], [`SL:ProtectedCharacteristic`, *black*], . . .]. Due to the nature of our domain, where

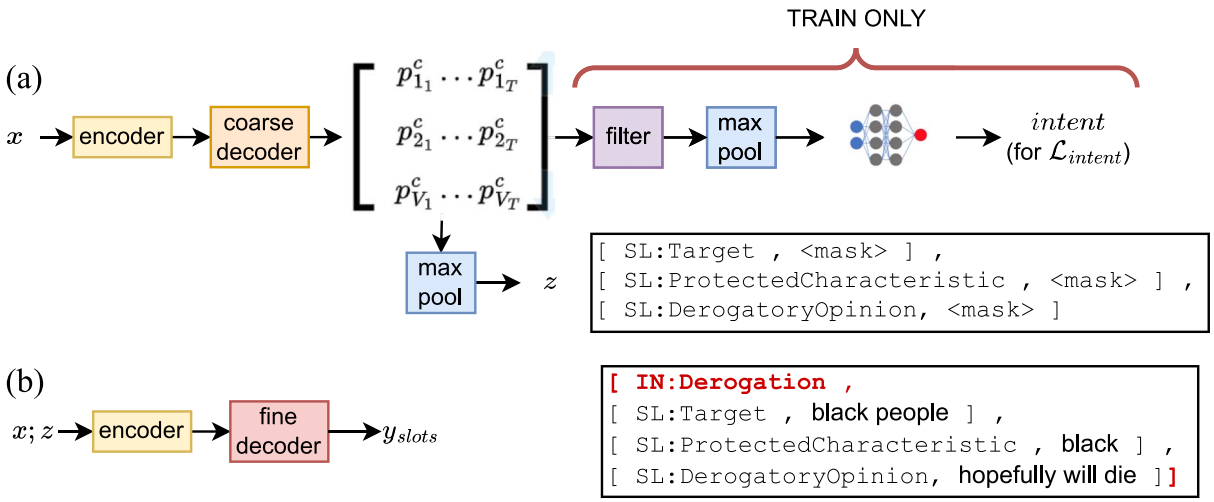

Figure 2: We first generate the meaning sketch based on the input post (a), and then refine it by filling the slots (b). The intent (in red) is inferred deterministically based on predicted slots y_{slots}. The model is trained with an intent-aware loss (a).

posts can contain multiple sentences, all of which might have to be considered to discover policy violations (e.g., because of coreference), we adopt the *conversational* approach to ICSF introduced in Aghajanyan et al. (2020). In this setting, all sentences are parsed in a single session (rather than utterance-by-utterance) which is pertinent to our task, as we infer intent *after* filling the slots, and would otherwise have no information on which slots to carry over (e.g., detecting a target in the first utterance does not constrain the set of slots that could occur in the following ones).

Our sequence-to-sequence model is built on top of BART (Lewis et al., 2020). However, in canonical ICSF, BART generates the intent first, and then uses it to look for the slots associated with it. In our case, intent is inferred post-hoc, based on the identified slots, not vice versa. Our model adopts a two-step approach where BART first generates a coarse representation of the input, namely, a meaning sketch with coarse-grained slots, and then refines it (Dong and Lapata, 2018). The meaning sketch is a tree where non-terminal nodes are slots, and leaves are `<mask>` tokens. The sketch for the example in Figure 1 and its refined version are shown in Figure 2. Specifically, we first encode source tokens w_i (Figure 2a):

$$e_1, \ldots, e_{|x|} = \text{Encoder}(w_1, \ldots, w_{|p|})$$

where $|x|$ is the number of tokens in post x, and then use the hidden states to generate the meaning sketch by computing probability distribution p^c over the vocabulary for each time step t as:

$$d_t^c = \text{Decoder}_c(e_1, \ldots, e_{|x|}; d_{t-1}^c; s_{t-1}^c)$$
$$\mathrm{p}_t^c = \text{softmax}(W d_t^c + b)$$

where s_{t-i}^c is the incremental state of the decoder. We then decode the meaning sketch $z_1, \ldots, z_T$:

$$z_t = \text{argmax}(\mathrm{p}_t^c)$$

And refine it (see Figure 2b) by first re-encoding the source tokens jointly with the meaning sketch (which is gold at training time, predicted otherwise):

$$v_1, \ldots, v_{|x|+T} = \text{Encoder}(w_1, \ldots, w_{|x|}; z_1, \ldots, z_T)$$

A second decoder generates then a new probability distribution over the vocabulary:

$$d_t^f = \text{Decoder}_f(v_1, \ldots, v_{|x|+T}; d_{t-1}^f; s_{t-1}^f)$$
$$\mathrm{p}_t^f = \text{softmax}(W d_t^f + b)$$

At inference time, we use beam search to generate the final representation starting from p^f.

The training objective is to jointly learn to generate the correct sketch z for post x, and the

correct tree t from x *and* z. We define our loss function for tuple (x, z, t) as:

$$L_{c,i} = -\sum_{v \in V} \mathbb{1}_{[z_i = v]} \ \log(\mathrm{p}^c_{i,v})$$
$$L_{f,i} = -\sum_{v \in V} \mathbb{1}_{[t_i = v]} \ \log(\mathrm{p}^f_{i,v})$$
$$\mathfrak{L} = \mathrm{mean}_i(L_{c,i}) + \mathrm{mean}_i(L_{f,i})$$

where V is the vocabulary and i is an index over the sequence length.

Although this loss penalizes the model for hallucinating or missing slots, it does not discriminate between errors that cause the prediction of a wrong intent, and those that are less relevant (e.g., hallucinating a threat when no target has been detected). In fact, intent is not part of our sequence-to-sequence task since it is only predicted post-hoc. To help the model learn how combinations of slots relate to intents, we include intent classification as an additional training task.

We essentially predict intent starting from the probability of each slot to appear in the sketch (Figure 2a). In other words, we restrict p^c_t to slot tokens (e.g., `SL:Target`) and normalize it, to obtain a new probability distribution q_t over the set of slots. We aggregate these probabilities by taking the maximum value over the sequence length, thus obtaining a single score for each slot. Since each intent can be modeled as a disjunction of slot combinations (e.g., the `NotHateful` intent could result from a tree containing only a target, or only a target AND a protected characteristic), we pass the slot scores through two linear layers with activation functions:

$$s_{slots} = \mathrm{ReLU}(W_{s2s}\,\mathrm{q} + b_{s2s})$$
$$s_{intent} = \mathrm{softmax}(W_{s2i}\,\mathrm{q} + b_{s2i})$$

thus obtaining a probability distribution s_{intent} over intents I. $W_{s2s} \in \mathbb{R}^{|S|\times|S|}$ models slot-to-slot interactions, while $W_{s2i} \in \mathbb{R}^{|S|\times|I|}$ models interactions between combinations of slots and intents. We then modify our loss to include the new classification loss for an input post with intent c:

$$L_{intent} = -\sum_{i \in I} \mathbb{1}_{[c=i]} \ \log(s_{intent,i})$$
$$\mathfrak{L} = \mathrm{mean}_i(L_{c,i}) + \mathrm{mean}_i(L_{f,i}) + L_{intent}$$

The new loss aims to assign higher penalty to meaning sketches that lead to intent misclassification. The two linear layers are trained on gold intents and sketches. The layers are then added to the BART-base architecture while kept frozen, so that the model cannot modify its weights to ''cover up'' wrong sketches by still mapping them to the right intents. Note that this additional classification task is only meant to improve the quality of the generated sketches: Intent is added post-hoc in the output tree depending on the slots that have been detected (Figure 2a).

6 Experimental Results

We performed experiments on the PLEAD dataset (Section 4). Rather than learning complex structures with nested slots, we post-process an instance with T targets into T equivalent instances, one per target. Furthermore, we discarded instances with reclaimed identity terms as these are not taken into account by our current modeling of the policy, and are too infrequent ($< 0.01\%$). We split the dataset into training, validation, and test set (80%/10%/10%), keeping the same intent distribution over the splits.

6.1 Why Explainability?

Our first experiment provides empirical support for our hypothesis that classifiers trained on collections of abusive and non-abusive posts do not necessarily learn representations directly related to abusive speech. We would further argue that if a model performs well on the test set, it has not necessarily learned to detect abuse. For this experiment, we trained RoBERTa (Vidgen et al., 2021b) with five different random seeds, and obtained an F1-score of $\sim 80\%$ in the binary classification setting with a low standard deviation (see Table 4). We further examined the output of these five RoBERTa models using AAA (Calabrese et al., 2021) and HateCheck (Röttger et al., 2021). AAA stands for Adversarial Attacks against Abuse and is a metric that better captures a model's performance on hard-to-classify posts, by penalizing systems which are biased on low-level lexical features. It does so by adversarially modifying the test data (based on patterns found in the training data) to generate plausible test samples. HateCheck is a suite of functional tests for hate speech detection models.

Seed	F1	AAA	GI_N	GI_P	IND
1	79.04	52.27	58.57	49.05	72.31
2	79.04	54.10	61.43	54.76	50.77
3	80.45	56.70	52.86	75.71	56.92
4	80.74	47.18	34.29	62.86	56.92
5	80.74	35.69	21.43	23.33	52.31
Std	0.89	8.31	17.20	19.44	8.54

Table 4: Performance of RoBERTa on PLEAD (measured by F1 and AAA) and HateCheck functionality tests for neutral (GI_N) and positive (GI_P) group identifiers and attacks on individuals (IND).

Firstly, we observe high standard deviations across AAA-scores. Models obtained with seeds 4 and 5 have identical F1-scores, but a gap of 12 points on AAA, suggesting that they may be modeling different phenomena. HateCheck tests on group identifiers confirm this hypothesis, as the model trained with random seed 5 misclassifies most neutral (GI_N) or positive (GI_P) sentences containing group identifiers as hateful, while the model trained with seed 4 can distinguish between different contexts and recognises most positive sentences as not hateful. Likewise, the models obtained with seeds 1 and 2 have identical F1-scores, and also similar AAA-scores, but a 20-point gap on the test containing attacks on individuals (IND). This suggests that classifiers tend to model different phenomena (like the presence of group identifiers or violent speech) rather than policy violations and that similarities in terms of F1-score disguise important differences amongst models.

6.2 Model Evaluation

Since the output of our model is a parse tree, we represent it as set of productions and evaluate using F1 (Quirk et al., 2015) on: (a) the entire tree (PF1), (b) the top level (i.e., productions rooted in intent, $PF1_I$), and (c) the lower level (i.e., productions rooted in correctly detected slots, $PF1_L$). We also report exact match accuracy for the full tree (EMA_T).

We compare our model (BART+MS+I) to ablated versions of itself, including a BART model without meaning sketches or an intent-aware loss, and a variant with meaning sketches but no intent-aware loss (BART+MS). We also compare against two baselines which encode the input post with an LSTM or BERT, respectively, and then use a feed-forward neural network to predict which slot labels should be attached to each token (Weld et al., 2021). The LSTM was initialized with GloVe embeddings (Pennington et al., 2014). For BERT, we concatenate the hidden representation of each token to the embedding of the `CLS` token, and compute the slots associated to a word as the union of the slots predicted for the corresponding subwords. We enhance these baselines by modeling slot prediction as a multi-label classification task (i.e., one-vs-one) in line with Pawara et al. (2020). For each pair of slots $< s_1, s_2 >$, we introduce an output node and use gold label 1 (-1) if s_1 (s_2) is the right tag for the token, and 0 otherwise.

As an upper bound, we report F1 score by comparing the annotations of one crowdworker against the others. Recall that annotation of hateful posts was simplified by asking participants to look for specific slots; as a result, some scores are only available for non-hateful instances where annotators could select from all the slots.

Our results are summarized in Table 5 (scores are means over five runs; hyperparameter values can be found in our code documentation). Our model achieves a production F1 of 52.96%, outperforming all comparison models. When looking at the top level of the tree ($PF1_I$), model performance on hateful instances (H) is considerably inferior to non-hateful ones (NH). This is not surprising, since hateful instances can be represented with ~ 4 sketches while non-hateful ones are noiser and can present a larger number of slot combinations. Model performance at filling correctly detected slots for hateful and non-hateful instances is comparable (61.93% and 62.66%), approaching the human ceiling. EMA_T scores are slightly higher for the non-hateful class, but this is not unexpected since hateful trees all have at least three slots, while many non-hateful ones have only one (i.e., a target).

Our model achieves an F1 of 57.17% on intent classification. In the binary setting, F1 jumps to 74.84%, suggesting that some mistakes on intent classification are due to the model confusing different hateful intents. As with all other models in the literature, the AAA-score is just below random guessing (Calabrese et al., 2021). Overall, improvement with respect to baselines is significant for all metrics. We also observe that both sketches and our intent-aware loss have a

Model	Tree												Intent	Hateful?		
	PF1			$PF1_I$			$PF1_L$			EMA_T			F1	F1	AAA	Mean
	H	NH	All	H	NH	All	H	NH	All	H	NH	All				
Humans	—	61.52	—	—	81.23	—	59.90	55.80	59.29	—	21.59	—	—	—	—	—
LSTM	40.66	20.08	36.30	54.20	18.42	45.97	33.91	32.03	34.63	0.00	0.90	0.11	48.10	57.73	40.55	49.14
BERT	40.45	16.58	35.26	59.54	16.92	50.31	25.39	23.43	25.02	0.00	0.00	0.00	52.35	76.77	31.19	53.98
BART	37.57	57.26	41.45	37.73	74.13	45.12	47.82	54.89	49.15	15.51	2.73	5.95	51.78	62.89	54.83	58.86
BART + MS	55.13	38.49	51.92	58.55	33.52	53.77	62.83	61.67	**62.63**	13.64	13.48	13.60	55.07	69.46	47.00	58.23
BART + MS + I	56.94	36.21	**52.96**	62.79	28.61	**56.29**	61.93	62.66	62.04	13.79	14.16	**13.88**	**57.17**	74.84	45.83	**60.34**

Table 5: Production F1 on the full tree (PF1), top layer ($PF1_I$) and lower layer ($PF1_L$); exact match accuracy on full tree (EMA_T); F1 on intent classification; F1 and AAA on binary classification. All scores are computed on the test set and reported as percentages of hateful (H) and non-hateful (NH) instances, and their combination (All). Best performing models are boldfaced.

1.	Never turn your back on [a brown skinned mud monkey]$_{\text{DehumanComp}}$ They do whatever the [jews]$_{\text{Target, PC}}$ tell them to do
2.	I d keep my distance if [I]$_{\text{Target}}$ was a l3zv0z because we re coming for you Strong and hard Beware
3.	Yes we gotta keep those [foxes]$_{\text{Target}}$ out [they steal natural resources]$_{\text{DehumanComp}}$ like food at the expense of our pets
4.	More [refugees]$_{\text{Target}}$ Cool more the merrier There is literally no reason not to do this
5.	Children are better off not being adopted and raised by sexually active [couples → `women`]$_{\text{Target, PC}}$ in general [their sexual degeneracy will mess]$_{\text{DerogatoryOpinion}}$ up the children they raise but it won t show immediately
6.	Yes we gotta keep those [foxes → `apes`]$_{\text{Target}}$ out [they steal natural resources]$_{\text{DehumanComp}}$ like food at the expense of our pets

Table 6: Posts that are incorrectly parsed (but not necessarily incorrectly classified) by our model.

large impact on the quality of the generated trees, and the intent predictions based on them. $PF1_L$ scores for BART + MS are higher but these are computed on *correctly* detected slots; the proportion of correct slots detected by this model is worse than the full model (see $PF1_I$ for BART+MS vs. BART+MS+I).

6.3 Error Analysis

We sampled 50 instances from the test set, and manually reviewed the trees generated by the five variants of our model (one per random seed). Overall, we observe that error patterns are consistent among all variants. In posts containing multiple targets, a recurrent mistake is to link the hateful expression to the wrong target, especially if the mention of the correct target is implicit (see example 1 in Table 6).

We also see cases where the parsing is coherent to the selected target, but this prevents the model from detecting hateful messages towards a different target (e.g., *''l3zv0z''* in example 2). Some mistakes stem from difficulty in distinguishing `DerogatoryOpinion` from other slots, as in example 3 where the opinion is misclassified as a dehumanizing comparison. This is a reasonable mistake, as comparisons to criminals are considered dehumanizing according to the policy (and therefore annotation instructions) and are often annotated as `DehumanisingComparison` in the dataset. We also observe that for posts correctly identified as non-hateful, the model tends to miss out on protected characteristics even when they occur (example 4). The model also hallucinates values for slots due to stereotypes prominent in the dataset. In example 5, *''women''* is mistakenly generated as the target of a sentence about sexual promiscuity (of couples), and in example 6 the model hallucinates *''apes''* as the animal in the comparison. In future work, hallucinations could be addressed by explicitly constraining the decoder to the input post.

Finally, we analyzed the behavior of the model in AAA scenarios, and observed that it struggles with counter speech, as the negative stance is often expressed with a negative opinion about the proponent of the hateful opinion, and therefore tagged as `DerogatoryOpinion`. Adding words that correlate with the hateful class to non-hateful posts succeeds in misleading our model; non-hateful instances often differ from hateful ones by a slot, rendering distractors more effective. However, for the same reason, the addition of such words can also flip the label (e.g., adding *''#kill''* to a post containing a target and a protected characteristic), and the model is incorrectly penalized by AAA (which assumes the label remains the same).

7 Discussion

The overwhelming majority of approaches to detecting abusive language online are based on training supervised classifiers with labeled examples.

Classifiers are expected to learn what abuse is based on these examples alone. We depart from this approach, reformulate the problem as policy-aware abuse detection, and model the policy explicitly as an Intent Classification and Slot Filling task. Our experiments show that conventional black-box classifiers learn to model *one* of the phenomena represented in the dataset, but small changes such as different random initialization can lead the very same model to learn different ones. Our ICSF-based approach guides the model towards learning policy-relevant phenomena, and this can be demonstrated by the explainable predictions it produces.

We acknowledge that policies for hate speech, as most human developed guidelines, leave some room for subjective interpretation. For instance, moderators might disagree on whether a certain expression represents a dehumanizing comparison. However, the more detailed the policy is (e.g., by listing all possible types of comparisons), the less freedom moderators will have to make subjective judgments. The purpose of policies is to make decisions as objective as possible, and our new problem formulation shares the same goal.

While our model still makes errors, the proposed formulation allows us to precisely pinpoint where these errors occur and design appropriate mitigation strategies. This is in stark contrast with existing approaches, where instability is the consequence of spurious correlations in the data, it is hard to isolate errors and, consequently, mitigation strategies are often not grounded in human knowledge about abuse. For example, our analysis showed that our model can sometimes fail to generate the correct tree by mixing the targets and sentiments of multiple opinions. This suggests that it would be useful to have nested slots, for example, a derogatory opinion as the child of its corresponding target. This could also help the model learn the difference between derogatory opinions (nested within a target node) and negative stance (nested within an opinion node), facilitating the detection of counter speech examples. Introducing a slot for the proponent of an opinion could also help, as the model would then recognise when a hateful opinion is expresed by someone other than the author.

Finally, we would like to emphasize that our modeling approach is not policy-specific and could be adapted to other policies used in industry or academia. Our formulation of abuse detection and the resulting annotation are compatible with more than one dataset (e.g., Vidgen et al., 2021a) and could be easily modified—for example, by removing or adding intents and slots. Extending our approach to other policies would require additional annotation effort, however, this would also be the case in the vanilla classification setting if one were to use a different inventory of labels.

8 Conclusions

In this work we introduced the concept of policy-aware abuse detection which we argue allows to develop more interpretable models and yields high-quality annotations to learn from. Humans who agree on the interpretation of a post also agree on its classification label. Our new task requires models to produce human-readable explanations that are specific to the input post. To enable models to reason over the policy, we formalize the problem of abuse detection as an instance of ICSF where each policy guideline corresponds to an intent, and is associated with a specific set of slots. We collect and release an English dataset where posts are annotated with such slots, and design a new neural model by adapting and enhancing ICSF architectures to our domain. The result is a model which is more reliable than existing approaches, and more ‘‘rational’’ in its predictions and mistakes. In the future, we would like to investigate whether and how the explanations our model produces influence moderator decisions.

Acknowledgments

We would like to thank the participants of our pilot study: Alberto Testoni, Alessandro Steri, Amr Keleg, Angelo Calabrese, Atli Sigurgeirsson, Christina Ovezeik, Cristina Coppola, Eddie Ungless, Elisa Caneparo, Erika Pan, Francesca De Donno, Gautier Dagan, Giuseppe De Palma, Jamila Moraes, Jie Chi, Julie-Anne Meaney, Kristian Iliev, Laura Forcina, Luigi Berducci, Marco Fumero, Maria Luque Anguita, Mariachiara Sica, Melina Gutiérrez Hansen, Nikita Moghe, Sandrine Chausson, Sara Forcina, Verna Dankers, and Wendy Zheng. We are also grateful to Eddie Ungless, Matthias Lindemann,

Nikita Moghe, and Tom Sherborne for their valuable feedback. Finally, we thank the action editor and the anonymous reviewers for their helpful comments. This work was supported in part by Huawei and the UKRI Centre for Doctoral Training in Natural Language Processing, funded by the UKRI (grant EP/S022481/1) and the University of Edinburgh, School of Informatics. Lapata gratefully acknowledges the support of the UK Engineering and Physical Sciences Research Council (grant EP/W002876/1) and the European Research Council (award 681760).

References

Armen Aghajanyan, Jean Maillard, Akshat Shrivastava, Keith Diedrick, Michael Haeger, Haoran Li, Yashar Mehdad, Veselin Stoyanov, Anuj Kumar, Mike Lewis, and Sonal Gupta. 2020. Conversational semantic parsing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16–20, 2020*, pages 5026–5035. Association for Computational Linguistics. https://doi.org/10.18653/v1/2020.emnlp-main.408

Wasi Uddin Ahmad, Jianfeng Chi, Tu Le, Thomas Norton, Yuan Tian, and Kai-Wei Chang. 2021. Intent classification and slot filling for privacy policies. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1–6, 2021*, pages 4402–4417. Association for Computational Linguistics. https://doi.org/10.18653/v1/2021.acl-long.340

Esma Balkir, Isar Nejadgholi, Kathleen Fraser, and Svetlana Kiritchenko. 2022. Necessity and sufficiency for explaining text classifiers: A case study in hate speech detection. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2672–2686, Seattle, United States. Association for Computational Linguistics. https://doi.org/10.18653/v1/2022.naacl-main.192

Valerio Basile, Michael Fell, Tommaso Fornaciari, Dirk Hovy, Silviu Paun, Barbara Plank, Massimo Poesio, and Alexandra Uma. 2021. We need to consider disagreement in evaluation. In *1st Workshop on Benchmarking: Past, Present and Future*, pages 15–21. Association for Computational Linguistics. https://doi.org/10.18653/v1/2021.bppf-1.3

Or Biran and Courtenay Cotton. 2017. Explanation and justification in machine learning: A survey. In *IJCAI-17 Workshop on Explainable AI (XAI)*, number 1, pages 8–13.

Agostina Calabrese, Michele Bevilacqua, Björn Ross, Rocco Tripodi, and Roberto Navigli. 2021. AAA: Fair evaluation for abuse detection systems wanted. In *WebSci '21: 13th ACM Web Science Conference 2021, Virtual Event, United Kingdom, June 21–25, 2021*, pages 243–252. ACM. https://doi.org/10.1145/3447535.3462484

Oana-Maria Camburu, Brendan Shillingford, Pasquale Minervini, Thomas Lukasiewicz, and Phil Blunsom. 2020. Make up your mind! Adversarial generation of inconsistent natural language explanations. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4157–4165. https://doi.org/10.18653/v1/2020.acl-main.382

Aida Mostafazadeh Davani, Mark Díaz, and Vinodkumar Prabhakaran. 2022. Dealing with disagreements: Looking beyond the majority vote in subjective annotations. *Transactions of the Association for Computational Linguistics*, 10:92–110. https://doi.org/10.1162/tacl_a_00449

Lucas Dixon, John Li, Jeffrey Sorensen, Nithum Thain, and Lucy Vasserman. 2018. Measuring and mitigating unintended bias in text classification. In *Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society, AIES 2018, New Orleans, LA, USA, February 02–03, 2018*, pages 67–73. ACM. https://doi.org/10.1145/3278721.3278729

Li Dong and Mirella Lapata. 2018. Coarse-to-fine decoding for neural semantic parsing. In *Proceedings of the 56th Annual Meeting of*

the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15–20, 2018, Volume 1: Long Papers, pages 731–742. Association for Computational Linguistics. https://doi.org/10.18653/v1/P18-1068

Antigoni Maria Founta, Constantinos Djouvas, Despoina Chatzakou, Ilias Leontiadis, Jeremy Blackburn, Gianluca Stringhini, Athena Vakali, Michael Sirivianos, and Nicolas Kourtellis. 2018. Large scale crowdsourcing and characterization of twitter abusive behavior. In *Twelfth International AAAI Conference on Web and Social Media*.

Sonal Gupta, Rushin Shah, Mrinal Mohit, Anuj Kumar, and Mike Lewis. 2018. Semantic parsing for task oriented dialog using hierarchical representations. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 – November 4, 2018*, pages 2787–2792. Association for Computational Linguistics. https://doi.org/10.18653/v1/D18-1300

Brendan Kennedy, Xisen Jin, Aida Mostafazadeh Davani, Morteza Dehghani, and Xiang Ren. 2020. Contextualizing hate speech classifiers with post-hoc explanation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5–10, 2020*, pages 5435–5442. Association for Computational Linguistics. https://doi.org/10.18653/v1/2020.acl-main.483

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880. https://doi.org/10.18653/v1/2020.acl-main.703

Binny Mathew, Punyajoy Saha, Hardik Tharad, Subham Rajgaria, Prajwal Singhania, Suman Kalyan Maity, Pawan Goyal, and Animesh Mukherjee. 2019. Thou shalt not hate: Countering online hate speech. In *Proceedings of the International AAAI Conference on Web and Social Media*, volume 13, pages 369–380. https://doi.org/10.1609/icwsm.v13i01.3237

Mary L. McHugh. 2012. Interrater reliability: The kappa statistic. *Biochemia Medica*, 22(3):276–282. https://doi.org/10.11613/BM.2012.031

Pushkar Mishra, Helen Yannakoudakis, and Ekaterina Shutova. 2019. Tackling online abuse: A survey of automated abuse detection methods. *CoRR*, abs/1908.06024.

Marzieh Mozafari, Reza Farahbakhsh, and Noël Crespi. 2019. A BERT-based transfer learning approach for hate speech detection in online social media. In *Complex Networks and Their Applications VIII - Volume 1 Proceedings of the Eighth International Conference on Complex Networks and Their Applications COMPLEX NETWORKS 2019, Lisbon, Portugal, December 10–12, 2019*, volume 881 of *Studies in Computational Intelligence*, pages 928–940. Springer. https://doi.org/10.1007/978-3-030-36687-2_77

Nedjma Ousidhoum, Zizheng Lin, Hongming Zhang, Yangqiu Song, and Dit-Yan Yeung. 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 2019, Hong Kong, China, November 3–7, 2019*, pages 4674–4683. Association for Computational Linguistics. https://doi.org/10.18653/v1/D19-1474

Pornntiwa Pawara, Emmanuel Okafor, Marc Groefsema, Sheng He, Lambert R. B. Schomaker, and Marco A. Wiering. 2020. One-vs-one classification for deep neural networks. *Pattern Recognition*, 108:107528. https://doi.org/10.1016/j.patcog.2020.107528

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543. https://doi.org/10.3115/v1/D14-1162

Chris Quirk, Raymond J. Mooney, and Michel Galley. 2015. Language to code: Learning semantic parsers for if-this-then-that recipes. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, ACL 2015, July 26–31, 2015, Beijing, China, Volume 1: Long Papers*, pages 878–888. The Association for Computer Linguistics. https://doi.org/10.3115/v1/P15-1085

Björn Ross, Michael Rist, Guillermo Carbonell, Benjamin Cabrera, Nils Kurowsky, and Michael Wojatzki. 2016. Measuring the reliability of hate speech annotations: The case of the european refugee crisis. In *3rd Workshop on Natural Language Processing for Computer-Mediated Communication/Social Media*, pages 6–9. Ruhr-Universitat Bochum.

Paul Röttger, Bertie Vidgen, Dirk Hovy, and Janet Pierrehumbert. 2022. Two contrasting data annotation paradigms for subjective NLP tasks. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 175–190, Seattle, United States. Association for Computational Linguistics. https://doi.org/10.18653/v1/2022.naacl-main.13

Paul Röttger, Bertie Vidgen, Dong Nguyen, Zeerak Waseem, Helen Z. Margetts, and Janet B. Pierrehumbert. 2021. Hatecheck: Functional tests for hate speech detection models. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1–6, 2021*, pages 41–58. Association for Computational Linguistics. https://doi.org/10.18653/v1/2021.acl-long.4

Maarten Sap, Saadia Gabriel, Lianhui Qin, Dan Jurafsky, Noah A. Smith, and Yejin Choi. 2020. Social bias frames: Reasoning about social and power implications of language. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 5477–5490. Association for Computational Linguistics.

Sheikh Muhammad Sarwar, Dimitrina Zlatkova, Momchil Hardalov, Yoan Dinkov, Isabelle Augenstein, and Preslav Nakov. 2022. A neighborhood framework for resource-lean content flagging. *Transactions of the Association for Computational Linguistics*, 10:484–502. https://doi.org/10.1162/tacl_a_00472

Alexandra Uma, Tommaso Fornaciari, Dirk Hovy, Silviu Paun, Barbara Plank, and Massimo Poesio. 2021. Learning from disagreement: A survey. *Journal of Artificial Intelligence Research*, 72:1385–1470.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4–9, 2017, Long Beach, CA, USA*, pages 5998–6008.

Bertie Vidgen, Alex Harris, Dong Nguyen, Rebekah Tromble, Scott Hale, and Helen Margetts. 2019. Challenges and frontiers in abusive content detection. In *Proceedings of the Third Workshop on Abusive Language Online*, pages 80–93. https://doi.org/10.18653/v1/W19-3509

Bertie Vidgen, Dong Nguyen, Helen Margetts, Patricia Rossini, and Rebekah Tromble. 2021a. 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. https://doi.org/10.18653/v1/2021.naacl-main.182

Bertie Vidgen, Tristan Thrush, Zeerak Waseem, and Douwe Kiela. 2021b. Learning from the worst: Dynamically generated datasets to improve online hate detection. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1–6, 2021*, pages 1667–1682. Association for Computational Linguistics. https://doi.org/10.18653/v1/2021.acl-long.132

William Warner and Julia Hirschberg. 2012. Detecting hate speech on the world wide web. In *Proceedings of the Second Workshop on Language in Social Media*, pages 19–26.

Zeerak Waseem, Thomas Davidson, Dana Warmsley, and Ingmar Weber. 2017. Understanding abuse: A typology of abusive language detection subtasks. In *Proceedings of the First Workshop on Abusive Language Online, ALW @ACL 2017, Vancouver, BC, Canada, August 4, 2017*, pages 78–84. Association for Computational Linguistics. https://doi.org/10.18653/v1/W17-3012

Henry Weld, Xiaoqi Huang, Siqu Long, Josiah Poon, and Soyeon Caren Han. 2021. A survey of joint intent detection and slot-filling models in natural language understanding. *arXiv preprint arXiv:2101.08091*. https://doi.org/10.1145/3547138

Xi Ye and Greg Durrett. 2022. The unreliability of explanations in few-shot in-context learning. *arXiv preprint arXiv:2205.03401*.

Guanhua Zhang, Bing Bai, Junqi Zhang, Kun Bai, Conghui Zhu, and Tiejun Zhao. 2020. Demographics should not be the reason of toxicity: Mitigating discrimination in text classifications with instance weighting. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5–10, 2020*, pages 4134–4145. Association for Computational Linguistics. https://doi.org/10.18653/v1/2020.acl-main.380

Frederike Zufall, Marius Hamacher, Katharina Kloppenborg, and Torsten Zesch. 2020. A legal approach to hate speech: Operationalizing the EU's legal framework against the expression of hatred as an NLP task. *arXiv preprint arXiv:2004.03422*.