

ToxCCIn: Toxic Content Classification with Interpretability

Tong Xiang¹ Sean MacAvaney^{1,2} Eugene Yang¹ Nazli Goharian¹
¹ IR Lab, Georgetown University, USA ² University of Glasgow, UK
tx39@georgetown.edu sean.macavaney@glasgow.ac.uk
eugene@ir.cs.georgetown.edu nazli@ir.cs.georgetown.edu

Abstract

Despite the recent successes of transformer-based models in terms of effectiveness on a variety of tasks, their decisions often remain opaque to humans. Explanations are particularly important for tasks like offensive language or toxicity detection on social media because a manual appeal process is often in place to dispute automatically flagged content. In this work, we propose a technique to improve the interpretability of these models, based on a simple and powerful assumption: a post is at least as toxic as its most toxic span. We incorporate this assumption into transformer models by scoring a post based on the maximum toxicity of its spans and augmenting the training process to identify correct spans. We find this approach effective and can produce explanations that exceed the quality of those provided by Logistic Regression analysis (often regarded as a highly-interpretable model), according to a human study.

1 Introduction

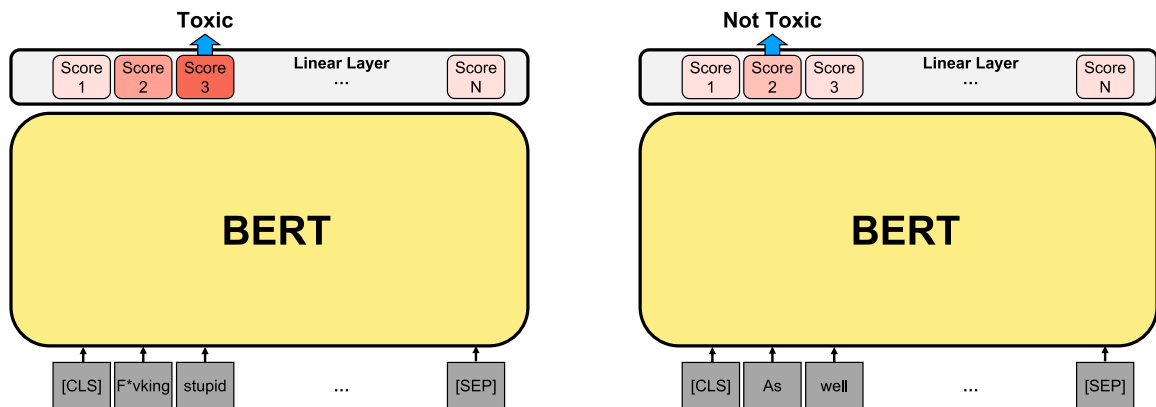
The rapidly increasing usage of social media has made communication easier but also has enabled users to spread questionable content (Matamoros-Fernández, 2017), which sometimes even leads to real-world crimes (Johnson et al., 2019; Committee et al., 2017; Center, 2017). To prevent this type of speech from jeopardizing others’ ability to express themselves in online communities, many platforms prohibit content that is considered abusive, hate speech, or more generally, toxic. To enforce such policies, some platforms employ automatic content moderation, which uses machine learning techniques to detect and flag violating content (Djuric et al., 2015; Nobata et al., 2016; MacAvaney et al., 2019).

Leveraging the development of pre-trained language models (Devlin et al., 2019; Liu et al., 2019b)

and domain transfer learning (Gururangan et al., 2020; Sotudeh et al., 2020), many models (Wiedemann et al., 2020) achieved high performance on toxicity detection (Zampieri et al., 2020). However, directly deploying such systems could be problematic for the following reasons. Despite being highly effective, one major problem is that the decisions of the systems are largely opaque, i.e., it can be difficult to reason why the model made its decision (Waseem and Hovy, 2016). This interpretability problem is especially apparent when compared to prior models, such as Logistic Regression, that transparently assign scores to each input feature (here, words) that can be used to justify the model’s decision. Knowing how one decision is made is important in toxicity detection. On one hand, recent laws such as General Data Protection Regulation¹ highlighted the significance of interpretable models for users; on the other hand, interpretable models can assist online community moderators in reducing their time spent on checking each potentially problematic post. Since the purpose of these explanations is for human consumption, we consider a model to be *interpretable* if it can produce a set of words from the input text that humans would consider a reasonable justification for the model’s decision.

In this work, we propose a technique to improve the interpretability of transformer-based models like BERT (Devlin et al., 2019) and ELECTRA (Clark et al., 2020) for the task of toxicity detection in social media posts. We base our technique on a simple and powerful assumption: *A post is at least as toxic as its most toxic span*. In other words, the toxicity of a piece of text should be associated with the most toxic span identified in the text. To this end, we propose using neural multi-task model that is trained on (1) toxicity detection over

¹<https://gdpr-info.eu/>



(a) When the input sequence is toxic, the toxicity of the most toxic span is picked to represent the toxicity of the sentence.

(b) When the input sequence is not toxic, none of the spans are toxic, and thus the whole sequence is predicted as non-toxic.

Figure 1: Illustration of our proposed approach. We showcase applying our proposed approach to the transformer-based model, BERT, in this case. In the linear layer, darker color denotes a more toxic span.

the entire piece of text and (2) toxic span detection (i.e., identifying individual tokens in the text that are toxic). Rather than the typical transformer classification approach, our model predicts the toxicity of each individual term in the text and aggregates them via max pooling to predict the toxicity of the entire text (see Fig. 1). Through experiments on the Civil Comment Dataset (Borkan et al., 2019), we find our proposed approach not only improves the classification effectiveness compared to models that are only trained on the classification task, but also helps when transferring the model to a similar task. More importantly, however, the structure of our model inherently generates explanations of the decision by selecting the terms with the highest toxicity scores. We find through a human study that these explanations exceed the quality of those provided by Logistic Regression—a model often regarded as highly interpretable. An error analysis has shown multiple insights into utilizing contextualized models in toxicity detection and lead future directions.

2 Related Work

Due to the ubiquity of online conversations, the need for automatic online toxicity detection has become crucial to promote healthy online discussions. Early research relied on surface-level features, e.g., bag-of-words approaches (Warner and Hirschberg, 2012; Waseem and Hovy, 2016; Davidson et al., 2017), and traditional machine learning methods. Although reported to be highly predictive (Schmidt and Wiegand, 2017) and easily interpretable, they

suffer from the problem of false positives as the presence of certain patterns could lead to misclassification (Kwok and Wang, 2013). For example, some slurs that frequently appear in African American English are usually picked as strong evidence of toxicity (Xia et al., 2020); these words are indeed innocuous and only confined within the black online community. These features also require such predictive terms to appear in both training and testing set to work effectively. Later on, neural textual representations have shown effectiveness in toxicity detection. Djuric et al. (2015) proposed using sentence-level embedding (Le and Mikolov, 2014) to represent the textual information and has shown great improvement against average over word-level embedding (Nobata et al., 2016). These representations are usually utilized together with either linear classifiers such as Logistic Regression (Djuric et al., 2015), or neural classifiers such as Convolutional Neural Networks (CNN) (Gambäck and Sikdar, 2017) and Long Short-Term Memory Networks (LSTM) (Badjatiya et al., 2017). More recently, large-scale pre-trained language models such as Bidirectional Encoder Representation from Transformer (BERT) (Devlin et al., 2019) have shown great advantages in toxicity detection (Zampieri et al., 2019b, 2020), by learning contextual word embeddings instead of static embeddings.

Despite the fact that systems can achieve astonishing performance on given datasets, they suffer from the problem of lacking interpretability. One way of providing interpretability is to explain predictions (Belinkov and Glass, 2019). For textual

data, such explanations could be those sub-strings (i.e., spans) that significantly influence the models’ judgments, which were named *rationales* in Zaidan et al. (2007). Zhang et al. (2016) proposed a CNN model that exploits document labels and associated rationales for text classification. On toxicity detection, while most of the works were done focusing on improving the model performances, less attention was paid to interpretability. SemEval-2021 Task 5 provided the Toxic Spans Detection Dataset (TSDD)² where each sample is annotated with both post-level label and rationales. Mathew et al. (2020) also provided a hate speech dataset where samples are annotated with rationales and other labels; though they also provided baseline models that can incorporate rationales, these models were built to evaluate the effectiveness of their proposed dataset, and thus cannot be easily transferred to other settings.

3 Model

We propose a neural multi-task model that can predict the toxicity and explain its prediction at the same time by providing a set of words that can justify its prediction. In this section, we introduce the assumption that empowers our model with interpretability. Then we present the proposed model’s architecture and the multi-task training paradigm.

3.1 Assumption

We begin with the following assumption: *A post is at least as toxic as its most toxic span.* This assumption suggests that if there is a word or phrase in a piece of text that is toxic, i.e., with a level of toxicity that is over a certain threshold, the toxicity level of the entire text is certainly over such threshold, and, therefore, should be considered toxic.

This assumption can be formalized as follows. Let $\mathbf{x} = \{x_1, x_2, \dots, x_n\}$ denote the input sequence where n is the length of the sequence. Given the input \mathbf{x} , we can define y as the toxicity label for the sequence and \mathbf{y} as the toxicity labels for individual token. Let $\mathbf{s} = \{s_1, s_2, \dots, s_n\}$ be a model’s prediction of toxicity for each token. By our assumption, we apply a max pooling operation over \mathbf{s} :

$$\tilde{s} = \max(\mathbf{s}) \quad (1)$$

where \tilde{s} represents the predicted toxicity of the entire sequence.

²<https://sites.google.com/view/toxicspans>

We acknowledge that this assumption may not always hold. In some cases, toxicity can be expressed in subtle or implicit ways, such as through sarcasm or metaphor (MacAvaney et al., 2019; Waseem et al., 2017). In such cases, there is often not a clearly identifiable span that is toxic. However, these cases are difficult for any model to identify, and through our experimental results in Section 5, we find that this does not hinder the effectiveness of our model.

3.2 Architecture & Methodology

To detect the toxicity and learn the toxic spans at the same time, we propose to use a neural multi-task learning framework (Caruana, 1997). In our settings, we jointly train the model with two related tasks: (1) Toxicity Detection (at the sequence level), and (2) Toxic Span Detection (at the token level). These two tasks share all the parameters in the model. Our approach can be applied to any sequence encoder model (e.g., LSTM or transformer). Given the input sequence \mathbf{x} , let the output of the sequence encoder be $\mathbf{H} = \{\mathbf{h}_1, \mathbf{h}_2, \dots, \mathbf{h}_n\}$, $\mathbf{H} \in \mathbb{R}^{n \times d}$. Here $\mathbf{h}_i \in \mathbb{R}^d$ denotes the i -th hidden state of the final layer, where d denotes the length of the hidden embedding. Unlike what Devlin et al. (2019) do for sequence classification where the linear layer is only stacked on the [CLS] token, we place it on top of the whole output sequence \mathbf{H} . Let $\mathbf{W} \in \mathbb{R}^{d \times 1}$ denote the parameters of the linear layer, then we have:

$$\mathbf{s} = \mathbf{H} \cdot \mathbf{W} \quad (2)$$

For the classification task, we use \tilde{s} as the predicted toxicity where \tilde{s} is calculated following the procedure mentioned in Eq. (1). For the span detection task, we directly leverage the output toxicity sequence \mathbf{s} . This setup ensures that the model learns to predict the text as toxic if a span is toxic.

For the purposes of training, let \mathcal{D}_1 be the dataset for toxicity detection task, and \mathcal{D}_2 be the dataset for toxic spans detection task. We construct the loss of the model \mathcal{L} to be the following:

$$\begin{aligned} \mathcal{L} = & \lambda \underbrace{\sum_{(\mathbf{x}, \mathbf{y}) \in \mathcal{D}_1} \mathcal{L}_C(\mathbf{x}, \mathbf{y})}_{\text{Loss for toxicity detection}} \\ & + (1 - \lambda) \underbrace{\sum_{(\mathbf{x}, \mathbf{y}) \in \mathcal{D}_2} \mathcal{L}_S(\mathbf{x}, \mathbf{y})}_{\text{Loss for toxic spans detection}} \end{aligned} \quad (3)$$

where \mathcal{L}_C is the loss for the toxicity detection task and \mathcal{L}_S is the loss for the toxic spans detection task. λ denotes a hyperparameter specifying the weight for each task. Here the toxicity detection task is a sequence classification task and the toxic spans detection task is a token classification task. We jointly train the model across tasks in an end-to-end fashion, minimizing the Mean Square Error (MSE) loss for both tasks.

4 Experiments

We conduct experiments to answer the following research questions:

- RQ1** Does our aforementioned assumption affect the model’s performance at detecting toxic content?
- RQ2** Is our approach applicable to different transformer models?
- RQ3** Does our approach produce models that can generalize to different domains?
- RQ4** Does our model identify spans that improve the interpretability of model decisions?

4.1 Data

We primarily train and evaluate our system using the Civil Comment Dataset (CCD) (Borkan et al., 2019). For interpretability, we leverage the Toxic Spans Detection Dataset (TSDD)² in a multi-task training paradigm. We also use the Offensive Language Identification Dataset (OLID) (Zampieri et al., 2019a) for the cross-domain evaluation.

CCD The CCD (Borkan et al., 2019) is a large-scale dataset with crowd-sourced post-level annotations for toxicity, provided by the Civil Comment platform.³ Posts that are rude, disrespectful, or unreasonable (Borkan et al., 2019) are considered toxic based on the rating guidelines as published by the Perspective API (Nithum et al., 2017; Wulczyn et al., 2017). Through the platform, crowd-sourced raters were asked to rate comments as “Very Toxic”, “Toxic”, “Hard to say”, or “Not Toxic”. A toxicity score between zero and one of a post is the fraction of raters considering it to be toxic. We further cast the scores to binary labels by setting a threshold of 0.5 (i.e., at least half of the raters consider the

post toxic). The dataset contains around 1.8 million posts in total, and 8% of them are labeled as toxic. **TSDD** TSDD² is a 10,000-sample subset of CCD, containing only toxic comments, marked up with individual spans that are toxic. Each post is annotated by three annotators. 528 posts have no annotated span since the annotators believe they are toxic as a whole without any explicit span.

OLID We use the OLID offensive language dataset (Zampieri et al., 2019b) to examine the domain-transferability. This dataset contains web posts with hierarchical annotation (Waseem et al., 2017). We use its first layer, where the annotations indicate whether the content is offensive (32%) or non-offensive (67%). For our evaluation, we use the official 860-sample OLID test set.

Curated CCD For the purpose of training and evaluation, we constructed a 30,000-sample curated CCD by mixing the clear-cut examples with the ambiguous ones, showcased in Table 1. Here, 14,000 samples are used for training and the rest are for testing.

We first sampled 7,000 highly toxic posts (toxicity score greater than 0.8) and 7,000 non-toxic posts (toxicity score less than 0.1) from CCD. Note that 3,000 of the toxic posts sampled were drawn from TSDD, which is still a subset of CDD, for the span annotations. These posts are considered to be *easy* since a great portion of the raters agreed on the judgment.

We further sampled another 8,000 ambiguous posts that have toxicity scores between 0.1 and 0.3 and contain terms that frequently appear in the toxic posts. Terms annotated at least 20 times as part of toxic spans in TSDD are considered frequent, resulting in a list of 62. The top 20 terms are presented in Table 2. We believe that these toxic terms are used in a non-toxic way in these posts, and, therefore, are good adversarial examples for the models to learn from the context instead of memorizing the frequent terms. To maintain an even proportion of toxic and non-toxic posts, we sample an additional 8,000 highly toxic posts.

4.2 Implementation

Here, we describe the implementation details of our baselines and the proposed models.

LR We use Logistic Regression (LR) classifier with lemmatized uni-gram and bi-gram features and L2 regularization as our baseline for both effectiveness evaluation and interpretability eval-

³The platform was shut down by the end of 2017: https://medium.com/@aja_15265/saying-goodbye-to-civil-comments-41859d3a2b1d

Comment	Toxicity	Label
Like me flagging your comments.	0	Non-toxic
We need to tax the clueless, irresponsible <u>idiots</u> .	0.898	Toxic
Don't take the bait of the <u>troll</u> . It's what they want.	0.167	Non-toxic

Table 1: Three samples from Curated CCD. We highlight the toxic spans in the toxic sample with purple underline. We mark the potentially toxic words in the non-toxic sample with pink double underline.

Word	Freq. ↓	Word	Freq. ↓
stupid	1085	idiotic	119
idiot	572	ridiculous	103
idiots	378	ass	102
ignorant	270	fools	100
stupidity	253	damn	100
dumb	185	racist	97
moron	163	loser	91
fool	163	morons	88
pathetic	145	hypocrite	65
crap	127	shit	62

Table 2: The 20 most frequent entries of the toxic word list for finding ambiguous non-toxic samples in CCD.

uation. LR is shown to be effective in toxicity detection (Djuric et al., 2015; Waseem and Hovy, 2016) while naturally providing explanations for its predictions if utilized together with bag-of-tokens features. While there are other models that rely on the attention mechanisms to provide various interpretability (Rogers et al., 2020), we choose to use LR instead of them. On the one hand, they usually need postprocessing to perform interpretation, which is less straightforward and intuitive compared to LR; on the other hand, there are still debates about whether attention mechanisms can produce meaningful explanations (Wiegrefe and Pinter, 2019; Jain and Wallace, 2019). For our LR baseline, NLTK is used for preprocessing and scikit-learn is used for building the classification model.

For the following transformer-based models, we utilize the transformers (Wolf et al., 2020) library with Adam (Kingma and Ba, 2015) optimizer. The learning rate is by default set to 2×10^{-5} and the number of training epochs is tuned on the validation set. We also utilized the pre-trained BERT-base (Devlin et al., 2019) and ELECTRA-base (Clark et al., 2020) that are available in the Huggingface community. Input sen-

tences are trimmed to a max length of 256 tokens. **BERT/ELECTRA-CLS** As baselines, we evaluate typical sentence classification (CLS) architectures tuned on only the post-level labels by adding a linear layer on top of the [CLS] token. Both BERT and ELECTRA are trained with the cross-entropy loss, which is the default setting in the transformers library.

BERT/ELECTRA-SP We also evaluate models that are only trained for the span detection (SP) task. It follows the architecture and methodology mentioned in Section 3.2, except only optimizing the toxic spans detection loss \mathcal{L}_S . Since only a portion of the samples contains toxic span annotations, the models are only trained on that subset. Beyond the toxic spans detection task, we also evaluate the toxicity detection performance for these models leveraging our proposed assumption, even though these models are not trained for toxicity detection.

BERT/ELECTRA-MT Finally, we describe the implementation of our proposed Multi-Task (MT) models. The models are built with the architecture described in Section 3.2 and optimized with the joint loss \mathcal{L} shown in Eq. (3). Since not all the input posts have the labels for toxic spans, the multi-task models are trained by interleaving samples with and without span information. Joint loss \mathcal{L} is calculated for those samples with labels for both tasks; for samples with only post-level labels, we calculate only the classification loss \mathcal{L}_C . During training, we interleave the update with each kind of loss to ensure a balance update on the parameters. We specify the hyperparameter λ to be 0.5 (i.e., weighting both tasks equally).

5 Results on Toxicity Detection

In this section, we answer the first three research questions by presenting and analyzing the classification effectiveness of the proposed models in various settings.

The in-domain toxicity detection performance is

Model	Non-Toxic			Toxic			Macro-F1
	P	R	F1	P	R	F1	
LR	0.824	0.828	0.826	0.827	0.823	0.825	0.826
BERT-CLS	0.944	0.674	0.786	0.746	0.960	0.840	0.813
ELECTRA-CLS	0.952	0.617	0.749	0.717	0.969	0.824	0.787
BERT-SP	0.843	0.771	0.806	0.789	0.857	0.822	0.814
ELECTRA-SP	0.884	0.716	0.791	0.761	0.906	0.827	0.809
BERT-MT	0.900	0.879	0.890	0.882	0.903	0.892	0.891
ELECTRA-MT	0.904	0.880	0.892	0.883	0.907	0.894	0.893

Table 3: In-domain evaluation results for toxicity classification on the Curated CCD dataset. We report Precision (P), Recall (R), and F1 for each model on all categories. We also report the Macro-F1 for all models. The best F1 performance is indicated in **bold**.

shown in Table 3. Both of our proposed multi-task models BERT-MT and ELECTRA-MT achieve the best performance among all models we evaluated. Models using the [CLS] tokens perform the worst among others, even worse than the ones that are only trained on the span detection task. This suggests that the span information is capable of predicting the toxicity of the entire content and combining it with the post-level supervised information further improves the effectiveness. This answers RQ1: our proposed model based on our assumption improves the classification effectiveness. Furthermore, BERT-MT and ELECTRA-MT are equally effective in respect to various evaluation metrics; therefore, we validate RQ2: our approach generalizes to at least two pre-trained transformer models.

Interestingly, LR has the highest precision among the baselines but the lowest recall, suggesting that the model may be relying on high-precision features such as racial slurs. This matches previous observations by MacAvaney et al. (2019).

In a domain transfer setting, i.e., training on CDD and testing on OLID, our approach is competitive with models trained on OLID. In Table 4, both BERT-MT and ELECTRA-MT obtain a competitive 0.77 macro-F1 without training on any example in OLID, even outperforming some leading systems reported on this dataset. Therefore, we confirm that our models remain effective in a domain-transfer setting (RQ3).

5.1 Error Analysis

To better understand the limitations of our proposed models, we qualitatively analyze the predictions against the gold labels using BERT-MT.

False Positives One major source of false positives comes from treating negative words, usually

Model	Macro-F1
SVM (Zampieri et al., 2019a)	0.69
BiLSTM (Zampieri et al., 2019a)	0.75
BERT-FT (Liu et al., 2019a)	0.83
BERT-MT (ours, transfer)	0.77
ELECTRA-MT (ours, transfer)	0.77

Table 4: Evaluation on the OLID. BERT-FT stands for the BERT model fine-tuned on the OLID data. Our system performs competitive in a completely transfer setting (only trained on CCD data). We report Macro-F1 for all models here. The best performance is in **bold**.

adjectives, such as *disgusting*, *lazy*, *incompetent*, as strong signals for toxicity. Also, our proposed models tend to treat certain sub-words that frequently appear in toxic contexts as toxic spans. For example, the sub-word *##nt* of *magnificent* is predicted as toxic (0.979⁴) in the following sentence:

Should have had the magnificent Doug Ford stump for Smith....LOL

This could be attributed to its high frequency in explicitly toxic words such as *ignorant* and *arrogant*.

We also suspect that the model is over-leveraging expression patterns or co-occurrences for toxicity classification. For example:

I have never seen a suspect identified as a "brown" man.

Here, the model misclassifies it as toxic and picks the word *brown* as the most toxic span with a toxicity score of 0.527. We first rule out the possibility that the toxicity comes from the negation

⁴Toxicity score.

by experimentally removing the word *never* from the sentence. We find that the toxicity score of the sentence increases to 0.762 with the word *brown* still as the most toxic span. We then examine by replacing the word *suspect* and keep the rest; when we replace *suspect* with *girl* or *mom*, the toxicity score for *brown* decreases to 0.334 and 0.141 respectively; when changing to *prisoner* or *scammer*, the toxicity score for word *brown* becomes 0.717 and 0.972. Among all the cases that we have experimented with, the word *brown* is consistently the most toxic word in the sentence. It seems that the model is learning the correlation between the noun *suspect* (or any other word positioned here) and the adjective *brown*. Though it might lead to false positives in rare cases, the predictive power that enables this phenomenon could potentially be the reason that our model is doing better in picking unseen words as toxic spans than the LR model since it infers the prediction without leveraging on the lexical information but the syntactic information.

False Negatives Some comments are inherently hard for both human and machine classifiers to identify. There are cases where the sentences can reasonably be considered as non-toxic where our model is also predicting as such. For example:

Ignore the trolls Sheema.....You are great and I always enjoy your pieces.

LOL....I keep hearing Garland never got a hearing blah,blah,blah....It sucks being in the minority....Go win some elections..

When the headline reads "Steve Bannon's porn and meth house"

In the first and second example, the speaker is not intended to be toxic even with the appearance of the word *trolls* and *sucks*. For the third example, the toxicity lies in the quoted text, which is also not the intention of the speaker. It is debatable that whether our model is indeed making mistakes on these naturally ambiguous comments.

Another major source of false negatives comes from the unawareness of the outside context, e.g., the target of the comment. For example:

Degenerate comment.

I'm beginning to think the left lacks the mental capacity to reason.

Model	SD-P	SD-R	SD-F1
LR	0.111	0.195	0.120
BERT-SP	0.836	0.798	0.792
ELECTRA-SP	0.840	0.807	0.798
BERT-MT	0.837	0.785	0.784
ELECTRA-MT	0.842	0.788	0.789

Table 5: Evaluation results on toxic spans detection task. The best performance is **bolded**.

These words/phrases such as *degenerate*, and *lacks the mental capacity to*, which could also be utilized in the neutral descriptions, are used here for expressing toxicity. It is easy for a human to reason out the target mentioned in the sentence and thus be aware of the toxicity raised; this is usually not the case for machine classifiers.

Also, many users intentionally modify the explicitly toxic words (e.g., replacing or removing characters) to obfuscate automatic detection while still keeping their intent clear to human (Djuric et al., 2015), e.g., changing word *idiots* into *I.d.i.o.t.s* or modifying word *asshole* to *a-hole*. Even for advanced transformer models, they still need to learn deeper information or be incorporated with more human supervision to be fully aware of these cases.

6 Interpretability

Finally, we examine the interpretability of our models. We first leverage the existing span annotations to evaluate but discovered that they limit our analysis. To overcome the limitation, we further conduct a user study to evaluate the interpretability directly.

6.1 Span Detection as Interpretation

We select a balanced 8,000-sample set from the test split of the Curated CCD. Here, non-toxic posts have toxicities between 0 and 0.1 and therefore are all considered to have no toxic spans; toxic samples are all from TSDD. We follow the ad-hoc evaluation metrics which are introduced in Da San Martino et al. (2019) and utilized in SemEval 2021 Task 5² – Span Detection Precision (SD-P), Recall (SD-R), and F1 (SD-F1). Due to the nature of the toxicity span detection task where instances span from single tokens to multiple sentences, the ad-hoc evaluation metrics give partial credits to imperfect matches at the character level. Given a post t , let the ground truth be a set of character offsets S_G^t and let one certain system A_i return a

set of character offsets $S_{A_i}^t$. With the system A_i and ground truth, the SD-P and SP-R on post t are then defined as follow:

$$P^t = \frac{|S_{A_i}^t \cap S_G^t|}{|S_{A_i}^t|} \quad (4)$$

$$R^t = \frac{|S_{A_i}^t \cap S_G^t|}{|S_G^t|}$$

With SD-P and SD-R defined, the SD-F1 is also defined:

$$F_1^t = \frac{2 \cdot P^t \cdot R^t}{P^t + R^t} \quad (5)$$

We report the average values over all samples. The evaluation results are shown in Table 5.

Models trained for the span detection task (*-SP) achieve the highest F1 scores with no surprise. However, our multi-task models provide nearly equal span detection effectiveness with far better performance on toxicity detection (See Table 3). Also, BERT-MT and ELECTRA-MT perform similarly here, further confirming that our approach can be used in various transformer models (RQ2).

We also compared with LR, which is widely considered to be interpretable. The predicted span from the LR model is reconstructed from the bag-of-token features. Features that contribute to the positive score for the sample are considered as the *interpretation* of the model. Our proposed models were shown to strongly outperform LR by a large margin. Note that LR’s performance is hindered in this evaluation, as it was not trained for sequence classification (e.g., what CRF would do), despite being considered an interpretable model.

6.2 User Study

The labeled toxic spans used for the evaluation in Section 6.1 are not annotated to be the *interpretation* for the toxicity of the post. We argue that it is not sufficient to evaluate the interpretability of prediction along with other known issues in automated evaluation (Manning et al., 2020; Novikova et al., 2017).

Therefore, we select 400 toxic samples from the test split of Curated CCD and 237 offensive samples from the testing set of OLID for the interpretability study. For each sample, three words with the highest predicted score from the models are picked as the explanation and are post-processed to the same form to avoid identifying the models based on the types of token preprocessing. Annotators are asked to annotate for the

toxicity (e.g., whether the sample is toxic) of the samples and pick the model with a better explanation. The order and the name of the models are hidden from the annotators to avoid biases. Although ELECTRA-MT and BERT-MT perform comparably in terms of F1, we found that qualitatively BERT-MT is better and therefore picked for the user study.

Despite the subjectivity of the annotation task, the annotators agreed in 78% of the cases. For our quantitative analysis, we use the samples that the annotators agree on. We also filter out 22 samples that both of the annotators considered non-toxic.

Preference	CCD	OLID	All
LR	26.5%	18.8%	24.0%
BERT-MT	59.4%	41.6%	53.7%
No Preference	14.2%	39.6%	22.3%

Table 6: Aggregated human preference during the interpretability experiment for CCD and OLID. We see that the annotators prefer the explanations from the BERT-MT over those from LR by a hefty margin.

On average, our annotators prefer BERT-MT over LR on more than half of the samples. As shown in Table 6, BERT-MT is considered to be more interpretable in both datasets by a wide margin. This result not only suggests that our proposed assumption provides interpretability to transformer models (RQ4) but also better explanations than the widely-known interpretable models.

We now take a closer look at how BERT-MT model’s explanations compare to those from LR. Some examples are shown in Table 7. We find that the predictions from BERT-MT are more polarized, while those from LR tend to be neutral. This is because the usage of MSE loss in BERT-MT penalizes values for not being close to 0 or 1; for LR, the L2 regularization penalizes large weight values. It is also worth noting that these models differ in how term scores are aggregated over the post; BERT-MT takes the maximum, whereas LR takes the sum. For analysis, we highlight three cases here:

When BERT-MT is preferred. BERT-MT can pick out the toxic spans more accurately by leveraging the context. In comparison, the LR model suffers from picking words based on frequency even with a non-toxic usage, resulting in a bias toward some entities, such as certain groups of people (See example A in Table 7). BERT-MT also takes advantage of the WordPiece tokenization which re-

#	Model	Example
A	BERT-MT	red #neck followers are doing all they can to make america white again .
	LR	redneck follower are doing all they can to make america white again .
B	BERT-MT	not only is he a liar , he ' s a coward too ! the clown is going down !
	LR	not only is he a liar , he ' s a coward too ! the clown is going down !
C	BERT-MT	wow you ' re un - smart .
	LR	wow you ' re un-smart .
D	BERT-MT	your constant ability at being an a - hole liar doesn ' t make you great
	LR	your constant ability at being an a-hole liar doe n't make you great

Table 7: Examples for qualitative analysis on interpretability. Here red color indicates toxicity, blue color indicates innocuousness, and white color represents neutral. Darker colors indicate more polarity.

tains the toxic word pieces when users are altering words to avoid censorship; while LR lemmatizes the tokens and resulting in losing this information. **When LR is preferred.** We observe that LR tends to predict more spans than BERT-MT, leading to higher recall, resulting in more *comprehensive* explanations. Take the example B in Table 7 for instance, the LR is able to pick out the word *coward* and *clown* which are considered to be non-toxic by BERT-MT. We conclude that BERT-MT is more cautious in predicting words as toxic.

When both models are equal. We find that both models are doing poorly if the sentence is implicitly toxic. Such indirect toxicity is carried either by negation (Example C in Table 7) or adversarially-modified toxic words (Example D in Table 7). Finally, there are also many cases where both models are doing equally well. This category of samples is generally easier to detect with explicit terms or slurs without ambiguity.

7 Conclusion & Future Work

In this paper, we proposed a toxicity detection approach that builds in the interpretability by predicting the toxicity of a piece of text based on the toxicity level of its spans. We showed that our approach is more effective in both in-domain and cross-domain evaluation than baselines that were shown to be effective. By conducting a user study, we further showed that our approach generates better explanations of the classification decisions than what Logistic Regression produces, which is known to be interpretable.

In the future, we plan to extend our current work

in several ways. We plan to take the implicit toxicity into consideration to make our assumption more robust. Besides, we will dig more into toxicity detection with long sequences. We also plan to investigate methods that consider more subtle context and actors in the content to better distinguish different usages of the toxic terms.

Acknowledgements

This work was supported in part by the ARCS Foundation.

References

- Pinkesh Badjatiya, Shashank Gupta, Manish Gupta, and Vasudeva Varma. 2017. [Deep learning for hate speech detection in tweets](#). In *Proceedings of the 26th International Conference on World Wide Web Companion*, WWW '17 Companion, page 759–760, Republic and Canton of Geneva, CHE. International World Wide Web Conferences Steering Committee.
- Yonatan Belinkov and James Glass. 2019. [Analysis methods in neural language processing: A survey](#). *Transactions of the Association for Computational Linguistics*, 7:49–72.
- Daniel Borkan, Lucas Dixon, Jeffrey Sorensen, Nithum Thain, and Lucy Vasserman. 2019. [Nuanced metrics for measuring unintended bias with real data for text classification](#). In *Companion Proceedings of The 2019 World Wide Web Conference*, WWW '19, page 491–500, New York, NY, USA. Association for Computing Machinery.
- Rich Caruana. 1997. Multitask learning. *Machine learning*, 28(1):41–75.
- Southern Poverty Law Center. 2017. The year in hate and extremism.

- Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. 2020. [Electra: Pre-training text encoders as discriminators rather than generators](#). In *International Conference on Learning Representations*.
- Home Affairs Committee et al. 2017. Hate crime: Abuse, hate and extremism online. *Fourteenth Report of Session 2016*, 17.
- Giovanni Da San Martino, Seunghak Yu, Alberto Barrón-Cedeño, Rostislav Petrov, and Preslav Nakov. 2019. [Fine-grained analysis of propaganda in news article](#). 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)*, pages 5636–5646, Hong Kong, China. Association for Computational Linguistics.
- Thomas Davidson, Dana Warmsley, Michael Macy, and Ingmar Weber. 2017. Automated hate speech detection and the problem of offensive language. In *Proceedings of the 11th International AAAI Conference on Web and Social Media, ICWSM '17*, pages 512–515.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. [BERT: Pre-training of deep bidirectional transformers for language understanding](#). In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.
- Nemanja Djuric, Jing Zhou, Robin Morris, Mihajlo Grbovic, Vladan Radosavljevic, and Narayan Bhamidipati. 2015. [Hate speech detection with comment embeddings](#). In *Proceedings of the 24th International Conference on World Wide Web, WWW '15 Companion*, page 29–30, New York, NY, USA. Association for Computing Machinery.
- Björn Gambäck and Utpal Kumar Sikdar. 2017. [Using convolutional neural networks to classify hate-speech](#). In *Proceedings of the First Workshop on Abusive Language Online*, pages 85–90, Vancouver, BC, Canada. Association for Computational Linguistics.
- Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A. Smith. 2020. [Don't stop pretraining: Adapt language models to domains and tasks](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8342–8360, Online. Association for Computational Linguistics.
- Sarthak Jain and Byron C. Wallace. 2019. [Attention is not Explanation](#). In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3543–3556, Minneapolis, Minnesota. Association for Computational Linguistics.
- NF Johnson, R Leahy, N Johnson Restrepo, N Velasquez, M Zheng, P Manrique, P Devkota, and Stefan Wuchty. 2019. Hidden resilience and adaptive dynamics of the global online hate ecology. *Nature*, 573(7773):261–265.
- Diederik P. Kingma and Jimmy Ba. 2015. [Adam: A method for stochastic optimization](#). In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.
- Irene Kwok and Yuzhou Wang. 2013. Locate the hate: Detecting tweets against blacks. In *Proceedings of the Twenty-Seventh AAAI Conference on Artificial Intelligence, AAAI'13*, page 1621–1622. AAAI Press.
- Quoc Le and Tomas Mikolov. 2014. Distributed representations of sentences and documents. In *Proceedings of the 31st International Conference on International Conference on Machine Learning - Volume 32, ICML'14*, page II–1188–II–1196. JMLR.org.
- Ping Liu, Wen Li, and Liang Zou. 2019a. [NULI at SemEval-2019 task 6: Transfer learning for offensive language detection using bidirectional transformers](#). In *Proceedings of the 13th International Workshop on Semantic Evaluation*, pages 87–91, Minneapolis, Minnesota, USA. Association for Computational Linguistics.
- Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019b. [Roberta: A robustly optimized bert pretraining approach](#).
- Sean MacAvaney, Hao-Ren Yao, Eugene Yang, Katina Russell, Nazli Goharian, and Ophir Frieder. 2019. [Hate speech detection: Challenges and solutions](#). *PLoS ONE*, 14:1–16.
- Emma Manning, Shira Wein, and Nathan Schneider. 2020. [A human evaluation of AMR-to-English generation systems](#). In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 4773–4786, Barcelona, Spain (Online). International Committee on Computational Linguistics.
- Ariadna Matamoros-Fernández. 2017. [Platformed racism: the mediation and circulation of an Australian race-based controversy on twitter, facebook and youtube](#). *Information, Communication & Society*, 20(6):930–946.
- Binny Mathew, Punyajoy Saha, Seid Muhie Yimam, Chris Biemann, P. Goyal, and Animesh Mukherjee. 2020. [Hatexplain: A benchmark dataset for explainable hate speech detection](#). *ArXiv*, abs/2012.10289.

- Thain Nithum, Dixon Lucas, and Wulczyn Ellery. 2017. Wikipedia talk labels: Toxicity. 1638–1644, Barcelona (online). International Committee for Computational Linguistics.
- Chikashi Nobata, Joel Tetreault, Achint Thomas, Yashar Mehdad, and Yi Chang. 2016. [Abusive language detection in online user content](#). In *Proceedings of the 25th International Conference on World Wide Web, WWW '16*, page 145–153, Republic and Canton of Geneva, CHE. International World Wide Web Conferences Steering Committee.
- Jekaterina Novikova, Ondřej Dušek, Amanda Cercas Curry, and Verena Rieser. 2017. [Why we need new evaluation metrics for NLG](#). In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2241–2252, Copenhagen, Denmark. Association for Computational Linguistics.
- Anna Rogers, Olga Kovaleva, and Anna Rumshisky. 2020. [A primer in BERTology: What we know about how BERT works](#). *Transactions of the Association for Computational Linguistics*, 8:842–866.
- 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.
- Sajad Sotudeh, Tong Xiang, Hao-Ren Yao, Sean MacAvaney, Eugene Yang, Nazli Goharian, and Ophir Frieder. 2020. [GUIR at SemEval-2020 task 12: Domain-tuned contextualized models for offensive language detection](#). In *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, pages 1555–1561, Barcelona (online). International Committee for Computational Linguistics.
- 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, Montréal, Canada. Association for Computational Linguistics.
- Zeeraq Waseem, Thomas Davidson, Dana Warmusley, and Ingmar Weber. 2017. [Understanding abuse: A typology of abusive language detection subtasks](#). In *Proceedings of the First Workshop on Abusive Language Online*, pages 78–84, Vancouver, BC, Canada. Association for Computational Linguistics.
- Zeeraq Waseem and Dirk Hovy. 2016. [Hateful symbols or hateful people? predictive features for hate speech detection on Twitter](#). In *Proceedings of the NAACL Student Research Workshop*, pages 88–93, San Diego, California. Association for Computational Linguistics.
- Gregor Wiedemann, Seid Muhie Yimam, and Chris Biemann. 2020. [UHH-LT at SemEval-2020 task 12: Fine-tuning of pre-trained transformer networks for offensive language detection](#). In *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, pages 1638–1644, Barcelona (online). International Committee for Computational Linguistics.
- Sarah Wiegrefe and Yuval Pinter. 2019. [Attention is not not explanation](#). 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)*, pages 11–20, Hong Kong, China. Association for Computational Linguistics.
- Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi 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 M. 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.
- Ellery Wulczyn, Nithum Thain, and Lucas Dixon. 2017. [Ex machina: Personal attacks seen at scale](#). In *Proceedings of the 26th International Conference on World Wide Web, WWW '17*, page 1391–1399, Republic and Canton of Geneva, CHE. International World Wide Web Conferences Steering Committee.
- Mengzhou Xia, Anjalie Field, and Yulia Tsvetkov. 2020. [Demoting racial bias in hate speech detection](#). In *Proceedings of the Eighth International Workshop on Natural Language Processing for Social Media*, pages 7–14, Online. Association for Computational Linguistics.
- Omar Zaidan, Jason Eisner, and Christine Piatko. 2007. [Using “annotator rationales” to improve machine learning for text categorization](#). In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Proceedings of the Main Conference*, pages 260–267, Rochester, New York. Association for Computational Linguistics.
- Marcos Zampieri, Shervin Malmasi, Preslav Nakov, Sara Rosenthal, Noura Farra, and Ritesh Kumar. 2019a. [Predicting the type and target of offensive posts in social media](#). In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1415–1420, Minneapolis, Minnesota. Association for Computational Linguistics.
- Marcos Zampieri, Shervin Malmasi, Preslav Nakov, Sara Rosenthal, Noura Farra, and Ritesh Kumar. 2019b. [SemEval-2019 task 6: Identifying and categorizing offensive language in social media \(OffensEval\)](#). In *Proceedings of the 13th International Workshop on Semantic Evaluation*, pages 75–86, Minneapolis, Minnesota, USA. Association for Computational Linguistics.

Marcos Zampieri, Preslav Nakov, Sara Rosenthal, Pepa Atanasova, Georgi Karadzhov, Hamdy Mubarak, Leon Derczynski, Zeses Pitenis, and Çağrı Çöltekin. 2020. [SemEval-2020 task 12: Multilingual offensive language identification in social media \(OffenseEval 2020\)](#). In *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, pages 1425–1447, Barcelona (online). International Committee for Computational Linguistics.

Ye Zhang, Iain Marshall, and Byron C. Wallace. 2016. [Rationale-augmented convolutional neural networks for text classification](#). In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 795–804, Austin, Texas. Association for Computational Linguistics.