ToXCL: A Unified Framework for Toxic Speech Detection and Explanation

Anonymous ACL submission

Abstract

The proliferation of online toxic speech is a pertinent problem posing threats to demographic groups. While explicit toxic speech contains offensive lexical signals, implicit one consists of coded or indirect language. Therefore, it is crucial for models not only to detect implicit toxic speech but also to explain its toxicity. This draws a unique need for unified frameworks that can effectively detect and explain implicit toxic speech. Prior works mainly formulated the task of toxic speech detection and explanation as a text generation problem. Nonetheless, models trained using this strategy can be prone to suffer from the consequent error propagation problem. Moreover, our experiments reveal that the detection results of such models are much lower than those that focus only on the detection task. To bridge these gaps, we introduce **ToXCL**, a unified framework for the detection and explanation of implicit toxic speech. Our model consists of three modules: a *(i) Target Group Generator* to generate the targeted demographic group(s) of a given post; an *(ii) Encoder-Decoder Model* in which the encoder focuses on detecting implicit toxic speech and is boosted by a *(iii) Teacher Classifier* via knowledge distillation, and the decoder generates the necessary explanation. **ToXCL** achieves new state-of-the-art effectiveness, and outperforms baselines significantly.

1 Introduction

Warning: *This paper discusses and contains content that can be offensive or upsetting.*

While social media has dramatically expanded democratic participation in public discourse, they have also become a widely recognized platform for the dissemination of toxic speech (Mathew et al., 2021; ElSherief et al., 2021; Yu et al., 2022). Online toxic speech, therefore, is prevalent and can lead the victims to serious consequences (Olteanu et al., 2018; Farrell et al., 2019). For this reason,

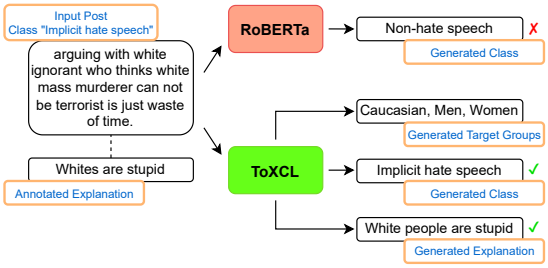

Figure 1: A sample input post and its ground truth explanation from the test set of `Implicit Hate Corpus` (ElSherief et al., 2021). The input post is fed into both models. The baseline model, RoBERTa, fails to detect the implicit toxic speech while our proposed **ToXCL** model successfully detects it and generates a toxic explanation that closely matches the ground truth explanation.

the development of toxic detection tools has received growing attention in recent years (Hutto and Gilbert, 2014; Ribeiro et al., 2018; Balkir et al., 2022; Jahan and Oussalah, 2023).

Toxic speech can generally be categorized as either *explicit* or *implicit*. Explicit toxic speech contains direct offensive language targeting individuals or groups (Nockleyby, 2000) and has been extensively studied (Schmidt and Wiegand, 2017; Jahan and Oussalah, 2021). On the other hand, implicit toxic speech presents a more challenging detection task as it relies on stereotypes and indirect language (ElSherief et al., 2021) (see Fig. 1) and has received limited attention. Moreover, given the absence of explicit offensive words or cues, it is crucial for AI models not only to detect implicit toxic speech but also to provide explanations for its toxic nature (Sridhar and Yang, 2022). The act of explanation serves practical purposes in various real-life applications, including improving human-machine interactions and building trustworthy AI systems (Ribeiro et al., 2016; Dosilovic et al., 2018; Bai et al., 2022).

These applications, therefore, pose a need for unified systems that can effectively detect implicit toxic speech and explain its toxicity. However,

previous works have mainly focused on a hybrid approach that combines detection and explanation tasks into a single text generation problem. For example, Sap et al. (2020) proposed concatenating the toxic speech label and explanation as the target output, AlKhamissi et al. (2022) and Huang et al. (2022) extended this approach by incorporating additional data such as target group(s) or social norms. Unfortunately, these hybrid approaches can introduce error propagation problems (Wu et al., 2018), possibly due to differences in training objectives (see Sec. 3.2.3). Consequently, models formulated in this manner tend to have much lower detection scores compared to models that focus solely on detection, as evidenced by our experimentation results (Sec. 4). Another simple approach is building a modular-based system separating the detection module and the explanation generation module. However, in reality, this kind of framework is computationally expensive to train, store and deploy as it has multiple components.

To bridge these gaps in detecting and explaining implicit toxic speech, we propose a unified framework **ToXCL** consisting of three modules (Fig. 2). Our approach is motivated by the findings that modeling the minority target groups associated with toxic speech can potentially improve the performance of both implicit toxic detection and explanation tasks (ElSherief et al., 2018; AlKhamissi et al., 2022; Huang et al., 2022). To achieve this, we build a *Target Group Generator* as our first module to generate the target minority group(s) based on the input post. The generated target group(s) and the post are then input into an *Encoder-Decoder Model* whose encoder detects the speech, and the decoder outputs the necessary toxic explanation. To enhance the detection performance of our encoder, we incorporate a strong *Teacher Classifier* that utilizes the *teacher forcing* technique during training to distill knowledge to our encoder classifier. Finally, we introduce a *Conditional Decoding Constraint* to enhance the explanation ability of the decoder during inference. Our model achieves state-of-the-art performance on the `Implicit Hate Corpus` (IHC) (ElSherief et al., 2021) and `Social Bias Inference Corpus` (SBIC) (Sap et al., 2020) in the task of implicit hate speech detection and explanation, outperforming baselines. Our contributions are as follows:

(i) We present a unified framework for the detection and explanation of implicit toxic speech. To the best of our knowledge, our work represents a pioneering effort in integrating both tasks into an encoder-decoder model to avoid the error propagation problem while maintaining the competitive performance on both tasks parameter-efficiently.

(ii) We propose to generate target groups for the toxic speech detection and explanation model through the utilization of an encoder-decoder model, thereby distinguishing our approach from previous methods (see Sec. 3.2.1). We also introduce several techniques to enhance the performance of our model: (1) joint training among the tasks to make the detection and explanation model end-to-end; (2) using teacher forcing to train the encoder; (3) a simple Conditional Decoding Constraint during the inference to avoid generating unnecessary explanation.

(iii) We set up new strong state-of-the-art results in the task of implicit toxic speech detection and explanation tasks in two widely used benchmarks `Implicit Hate Corpus` (IHC) and `Social Bias Inference Corpus` (SBIC).

(iv) We conduct a thorough analysis to demonstrate the effectiveness of our architectural design. We will open-source our model to inspire and facilitate future research.

2 Related Work

2.1 Toxic Speech Detection & Explanation

Early studies on identifying toxic speech relied on linguistic rule-based approaches (Chen et al., 2012; Hutto and Gilbert, 2014; Gitari et al., 2015; Wiegand et al., 2018). However, these methods, which use predetermined lexical and syntactic rules, struggle to detect implicit toxic speech without explicit vulgarities (Breitfeller et al., 2019; MacAvaney et al., 2019). Recent frameworks based on transformer architecture (Vaswani et al., 2017) have made progress in detecting toxic speech (Basile et al., 2019; Tran et al., 2020; Kennedy et al., 2020). However, detecting implicit toxic speech remains challenging despite attempts to improve performance on this task (Vidgen et al., 2019; Caselli et al., 2020, 2021; Kim et al., 2022). The issue of explaining why a text is toxic has received even more limited attention, with some studies focusing solely on explaining implicit toxic speech (ElSherief et al., 2021; Sridhar and Yang, 2022). Another few studies have addressed both implicit toxic speech detection and explanation (Sap et al., 2020; AlKhamissi et al., 2022; Huang et al., 2022), often formulating them as text generation tasks, possibly

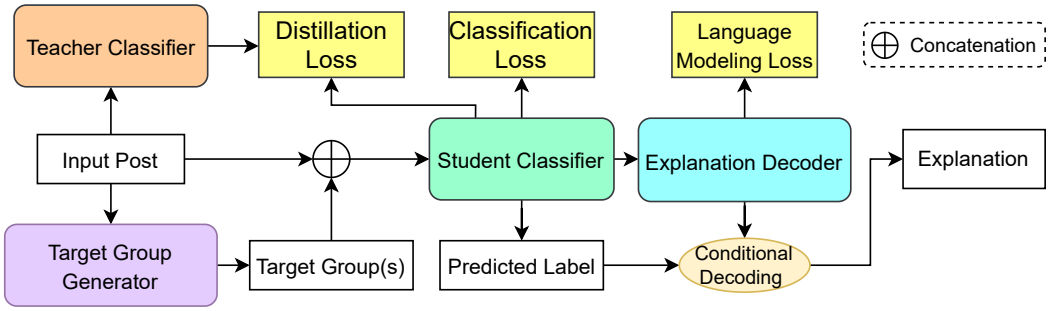

Figure 2: An overview of our proposed **ToXCL**. It consists of three modules: a *(i) Target Group Generator* generates the target group(s) of the input post; an *(ii) Encoder-Decoder Model* whose encoder focuses on implicit toxic speech detection whilst its decoder aims to generate necessary toxic explanation; a *(iii) Teacher Classifier* to distil the knowledge to the classifier encoder.

leading to error propagation and lower detection scores compared to detection-only models.

2.2 Knowledge Distillation

Knowledge distillation (Hinton et al., 2015) is a technique that enables a smaller student model to learn from a larger teacher model by transferring knowledge. It has proven effective in improving performance, reducing computational requirements, and increasing efficiency in the field of Computer Vision (Gou et al., 2021). Recently, researchers have explored applying knowledge distillation in Natural Language Processing. For example, Fu et al. (2020) used a contrastive approach to align the intermediate layer outputs of the teacher and student models. Turc et al. (2019) extensively studied the interaction between pre-training, distillation, and fine-tuning, demonstrating the effectiveness of pre-trained distillation in tasks like sentiment analysis. Additionally, Clark et al. (2019) trained a multitasking network by ensembling multiple single-task teachers. In our work, we distill the knowledge from a teacher classifier to our model's classifier (the student classifier), optimizing the Kullback-Leibler distance (Csiszár, 1975) between soft labels.

3 Methodology

3.1 Problem Formulation

The task of implicit toxic speech detection can be formulated as a binary classification problem while the explanation generation task can be regarded as a text generation problem. Each data instance $\langle IP, Y, E \rangle$ consists of an input post IP, a binary class label Y (1 for toxic speech, 0 for non-toxic speech), and a corresponding explanation E (`[None]` for non-toxic speech). The models then take IP as the input and learn to output Y and E.

3.2 ToXCL Framework

Figure 2 shows an overview of our proposed **ToXCL**, consisting of three modules: *(i) Target Group Generator*; *(ii) Encoder-Decoder Model*; *(iii) Teacher Classifier*. The details of each module are presented below.

3.2.1 Target Group Generator (TG)

Current toxic speech detection systems often overlook the nuances of toxic speech, which can be better addressed by modeling the minority target groups associated with it (ElSherief et al., 2018). Incorporating target group information has the potential to improve the accuracy of toxic speech detection and enable the generation of high-quality hate speech explanations (Huang et al., 2022). Therefore, we propose using a transformer-based encoder-decoder model (Raffel et al., 2020) to generate target minority groups for given posts, treating the task as a text generation problem rather than a multi-label classification task. This approach provides two advantages over classification models. Firstly, it leverages powerful pre-trained encoder-decoder models, enhancing the model's capabilities. Secondly, text generation models are more generalizable, as they are not restricted to a fixed number of target groups, allowing for greater flexibility in handling diverse scenarios.

After generating target groups G based on an input post IP, G and IP are concatenated as `"Target:{G} Post:{IP}"` and serve as the input for our **ToXCL**. The experimental details of the TG module are presented in Section 4.1.

3.2.2 Encoder-Decoder Model

Toxic speech detection and toxicity explanation are two tasks that have received increasing attention, and while researchers have made significant progress in separately solving each problem, ad-

dressing them together has received limited attention (Sap et al., 2020; AlKhamissi et al., 2022; Huang et al., 2022). However, these two tasks are strongly correlated, and the explanation of the post can potentially help the systems to detect toxic speech (AlKhamissi et al., 2022). Conversely, the toxicity explanation is sometimes only necessary when the post is detected as toxic. Typically, AlKhamissi et al. (2022); Huang et al. (2022) formulate both tasks as a single text generation task, which has some critical shortcomings as discussed in Section 1. Therefore, in this work, we propose a novel architectural design on top of a pre-trained encoder-decoder model. The encoder addresses the toxic speech detection task, while the decoder generates a toxicity explanation if the post is detected as toxic. The details of both the encoder and decoder components are introduced below.

• **Encoder Classifier (CL)** To enable the implicit toxicity detection capability, we build a binary classifier head on top of the encoder of a pre-trained encoder-decoder model. This head includes a linear layer that takes the average of token embeddings from the encoder's last hidden state as input, followed by a softmax layer (Goodfellow et al., 2016). To optimize the performance, both the encoder and the newly added classifier head are trained together using a binary cross-entropy loss:

$$\mathcal{L}_{cls} = -\frac{1}{N}\sum_{i}\sum_{j\in\{0,1\}} y_{ij}\log(p_{ij}) \quad (1)$$

in which $y_{i0}, y_{i1} \in \{0,1\}$, $p_{i0}, p_{i1} \in [0,1]$ and $p_{i0} + p_{i1} = 1$.

• **Explanation Decoder (ToX)** Recognizing the importance of generating explanations for implicit toxic speech and its potential impact on various applications, we utilize the decoder of our pre-trained encoder-decoder model to generate the explanation. To optimize its performance, the decoder is fine-tuned using a Causal Language Modeling (CLM) loss:

$$\mathcal{L}_{clm} = \sum_{i}\log(P(e_i|e_{i-k},...,e_{i-1};\theta)) \quad (2)$$

in which $E = \{e_1, e_2, ..., e_n\}$ is the set of tokens of the explanation, and k is the size of the window.

Finally, we train the encoder-decoder model for the task of toxic speech detection and explanation by joining the two losses:

$$\mathcal{L}_{xcl} = \alpha\mathcal{L}_{cls} + \beta\mathcal{L}_{clm} \quad (3)$$

in which α, β are the contribution weights.

3.2.3 Teacher Classifier (TC)

Since the open-sourced encoder-decoder models are commonly pre-trained on a diverse range of tasks and these tasks might not solely focus on learning strong representations from their encoders, these encoders may not exhibit the same strength as pre-trained encoder-based models such as BERT (Devlin et al., 2019) or RoBERTa (Liu et al., 2019) for classification tasks. Motivated by Hinton et al. (2015), we propose to use knowledge distillation to transfer knowledge from a strong encoder-based model (*Teacher Classifier*) to the Flan-T5 encoder (*Student Classifier*). Specifically, we leverage the outputs $\hat{y}tc$ and $\hat{y}sc$ from the *Teacher Classifier* and *Student Classifier*, respectively, and employ the Kullback-Leibler divergence loss (Csiszár, 1975) as the *teacher forcing* loss to minimize the discrepancy between $\hat{y}_{tc}$ and $\hat{y}_{sc}$:

$$\mathcal{L}_{tf} = D_{KL}(\hat{y}_{sc})||\hat{y}_{tc})) \quad (4)$$

Our final loss $\mathcal{L}$ is the weighted sum of $\mathcal{L}_{xcl}, \mathcal{L}_{tf}$:

$$\mathcal{L} = \lambda\mathcal{L}_{xcl} + \gamma\mathcal{L}_{tf} \quad (5)$$

in which λ, γ are the contribution weights.

3.2.4 Conditional Decoding Constraint (CD)

One of the main challenges with unified frameworks for toxic speech detection and explanation is the lack of synchronization between the classifier's output label and the generated explanation. For instance, when the classifier outputs a label of 0, indicating non-toxic speech, the explanation generation module still generates an explanation, even though it is unnecessary in this case. To address this, we propose the *Conditional Decoding Constraint*, a simple yet effective algorithm. This constraint controls the decoder's generation process by generating a `[None]` token for non-toxic speech and a toxic explanation for toxic speech. By incorporating this constraint, our framework ensures coherence and alignment between the generated explanations and classifier outputs, enhancing its overall performance.

4 Experimentation

4.1 Target Group Generator Experiment

Model	F1	ROUGE-L
BERT	68.35	70.44
GPT-2	53.98	56.86
BART	63.41	70.48
T5	**69.79**	**70.95**

Table 1: Target Group Generator experiments.

		IHC		SBIC	
Model	Size	Acc. (%)	F1 (%)	Acc. (%)	F1 (%)
RoBERTa-Large	354M	80.68	77.33	90.12	90.11
Teacher Classifier	354M	**82.52**	**79.49**	**91.19**	**91.19**

Table 2: Performance of Teacher Classifier, which is the RoBERTa-Large + **TG**.

• **Dataset** To address the problem of free-text target group labelling in IHC and SBIC datasets, we utilized the `HateXplain` dataset (Mathew et al., 2021), which provides 19 fine-grained categories for toxic speech. We fine-tune a T5 model (Raffel et al., 2020) as our TG model, and treat it as a text generation problem. To ensure our framework can predict specific target group(s) associated with posts from IHC and SBIC datasets, we conducted an analysis to identify any overlapping data between the `HateXplain` and IHC, SBIC datasets. We found only one instance of overlap, which we removed before training our TG model.

• **Baselines** We compare the performance of our TG model with three baseline models: (1) BERT (Devlin et al., 2019), an encoder-based model; (2) GPT-2 (Radford et al., 2019), a decoder-only model, (3) and BART (Lewis et al., 2020), an encoder-decoder model. BERT is widely used for multi-label classification tasks while both GPT-2 and BART have demonstrated remarkable performance in text

• **Implementation Details** We concatenate the annotated target group(s) in alphabetical order to construct the target label for each input post. All baselines and our TG model are initialized with pre-trained checkpoints from Huggingface (Wolf et al., 2020) and fine-tuned on a single Google CoLab P40 GPU with a window size of 256, a learning rate of $1e-5$, and AdamW (Loshchilov and Hutter, 2019) as the optimizer. The BERT model is fine-tuned for 10 epochs, while the GPT-2 and BART models are fine-tuned for 20k iterations. We use a beam search strategy with a beam size of 4 for our generation decoding strategy.

• **Automatic Evaluation** Our TG model is evaluated using F1 (%) for multi-label classification and ROUGE-L (%) (Lin, 2004). The results in Table 1 indicate that our model achieved an F1 score of 69.79 and a ROUGE-L score of 70.95, outperforming the competing baselines in identifying target groups in toxic posts.

4.2 Teacher Classifier Experiment

For our Teacher Classifier, a RoBERTa-Large model (Liu et al., 2019) is fine-tuned using the generated target group(s) (TG in Section 3). The model achieved an F1 score of 79.49 on IHC and 91.19 on SBIC, indicating the effectiveness of generated target group(s) in classifying toxic speech. Detailed results can be found in Table 2.

4.3 ToXCL Experiment

• **Dataset** We conduct our experiments on two datasets: IHC (ElSherief et al., 2021) and SBIC (Sap et al., 2020). These datasets are collected from popular social media platforms such as Twitter and Gab, providing comprehensive coverage of the most prevalent toxic groups. Prior to training, we pre-process the data as detailed in sec:preprocess-data.

• **Baselines** We compare **ToXCL** with three groups of baselines: *(G1) implicit toxic speech detection*, *(G2) implicit toxic speech explanation*, and *(G3) implicit toxic speech detection and explanation*.

For baselines in group G1, we use BERT (Devlin et al., 2019), HateBERT (Caselli et al., 2021), RoBERTa (Liu et al., 2019) and ELECTRA (Clark et al., 2020) as our baselines. They are widely employed in prior toxic speech detection works.

We select GPT-2 (Radford et al., 2019), BART (Lewis et al., 2020), T5 (Raffel et al., 2020), and Flan-T5 (Chung et al., 2022) as our baselines G2 and G3. GPT-2 represents the group of decoder-only models, while BART, T5, and Flan-T5 have the encoder-decoder architecture. Specifically for the models in group G2, we fine-tune them to generate `[None]` token or the explanations' tokens. For group G3, we concatenate the classification label `[Toxic]`/`[Non-toxic]` and the explanation of each sample as the output, and fine-tune the models with the post as input.

• **Implementation Details** We initialize all the models with the pre-trained checkpoints from Hug-

gingface (Wolf et al., 2020). We then fine-tune them on a single Google CoLab P40 GPU with a window size of 256, and a learning rate of $1e-5$ and use AdamW (Loshchilov and Hutter, 2019) as our optimizer. The classification baselines in group G1 are fine-tuned on 10 epochs while the generation ones in G2 and G3 are fine-tuned on 20k iterations. Beam search strategy with a beam size of 4 is utilized as our generation decoding strategy.

• **Automatic Evaluation** We adopt Accuracy and Macro F1 as our classification metrics, following prior works (Mathew et al., 2021; ElSherief et al., 2021). For the generation of explanations, we utilize BLEU-4 (Papineni et al., 2002), ROUGE-L (Lin, 2004) and METEOR (Banerjee and Lavie, 2005) as our n-gram metrics. We further utilize BERTScore (Zhang et al., 2020) to measure the similarity between the generated toxic explanation and the ground truth one based on deep-contextual embeddings. To ensure that unnecessary explanations are not generated for non-toxic posts and penalize unnecessary explanations, we develop a new evaluation algorithm for the explanation generation task. Its pseudo-code is presented in alg:geeration-scores. In this algorithm, we assign a score of 100 if both the generated explanation and the ground truth explanation are `[None]` indicating that no explanation is needed. If both the generated explanation and the ground truth explanation are not `[None]` we compute a score based on the quality of the generated explanation. For any other mismatched pairs, we assign a score of 0 to penalize the unnecessary explanations for non-toxic speech. It is worth noting that our evaluation algorithm is different from Sridhar and Yang (2022) which only evaluates the quality of the generation within implicit toxic cases.

• **Human Evaluation** To gain deeper insights into the *generation* performance and challenges that our **ToXCL** faces compared to the competing baseline, we conduct human evaluations using a randomly selected set of 150 samples that require explanations from each examined benchmark. Specifically, we collect the generated explanations from both the **ToXCL** and Flan-T5 models in two different settings, G2 and G3. To ensure high-quality evaluations, five native English speakers are hired to rate the generated explanations on a 1-3 scale (with 3 being the highest) based on three criteria: **(i) Correctness**, evaluating the accuracy of the explanation in correctly explaining the meaning of toxic speech; **(ii) Fluency**, assessing the fluency and coherence of the generated explanation in terms of language use; and **(iii) Toxicity**, gauging the level of harmfulness and judgmental tone exhibited in the generated explanation. The annotator agreement is measured using Krippendorff's alpha (Krippendorff, 2011), which provides a measure of inter-annotator reliability.

4.4 ToXCL Performance

• **Automatic Evaluation** Our experimental results in Table 3 reveal three main observations. Firstly, **ToXCL** outperforms all baselines on both benchmarks, demonstrating the effectiveness of our encoder-decoder model in addressing both implicit toxic detection and explanation tasks simultaneously without conflicts. Secondly, our model surpasses detection models in group G1, indicating the strong capability of our encoder in detecting implicit toxic speech. It is worth noting that despite having fewer parameters than the RoBERTa-Base model (124M), our encoder (Flan-T5's) classifier (109M) achieves better performance while maintaining computational efficiency. Lastly, our model significantly outperforms its backbone, Flan-T5, highlighting the effectiveness of our architectural designs in jointly training the tasks in an end-to-end manner for implicit toxic speech detection and explanation problems.

• **Human Evaluation** Our human evaluation results in Table 4 indicate that **ToXCL** outperforms its backbone model, Flan-T5, from groups G2 and G3, in terms of both explanation accuracy and textual fluency. This improved performance is also reflected in the detection task, resulting in more reliable explanations with fewer harmful outputs compared to the baselines. Our human annotators exhibit strong agreement with Krippendorff's alpha scores consistently measuring at least 0.78 among the three scores. It is worth noting that the average toxicity score of around 2 for both our **ToXCL** and baseline models aligns with expectations, given that the training datasets contain offensive words in the ground truth explanations (see Tab. 5).

5 Discussion

5.1 Case Study: Effectiveness of ToXCL

We compare the performance of **ToXCL** with the two best-performing baselines in group G3, namely GPT-2 on IHC and Flan-T5 on SBIC (Sec. 4). We

		IHC Detection		IHC Explanation				SBIC Detection		SBIC Explanation			
Model	Group	Acc.	Macro F1	BLEU-4	ROUGE-L	METEOR	BERTScore	Acc.	Macro F1	BLEU-4	ROUGE-L	METEOR	BERTScore
HateBERT	G1	78.67	75.93					89.32	89.31				
BERT		78.98	76.16					89.83	89.83				
ELECTRA		79.90	76.87					89.06	89.04				
RoBERTa		80.06	77.23					89.98	89.97				
GPT-2	G2			67.60	70.19	69.69	73.15			47.62	65.50	63.74	74.74
BART				53.67	59.18	57.40	68.14			45.22	67.73	67.25	83.98
T5				50.19	56.01	54.36	66.60			45.37	68.03	67.59	84.68
Flan-T5				47.33	53.78	51.95	65.14			45.83	68.37	67.98	85.04
GPT-2	G3	77.57	76.36	67.81	70.19	69.88	73.47	73.62	57.09	48.29	65.24	63.53	74.61
BART		70.40	64.11	55.92	60.48	58.87	68.74	88.25	88.20	45.65	68.08	67.81	84.35
T5		70.71	62.95	58.24	62.42	61.18	69.30	87.55	87.48	45.92	68.37	67.82	84.78
Flan-T5		71.52	65.56	56.58	61.75	60.20	69.84	88.40	88.37	45.98	68.81	67.99	85.07
GPT-2 + **TG**		74.00	58.12	67.86	70.38	69.71	73.11	78.87	77.93	48.56	65.48	63.65	75.80
BART + **TG**		74.28	58.88	67.75	68.17	69.68	73.67	88.55	88.52	45.70	68.41	67.81	85.03
T5 + **TG**		76.48	64.86	67.07	70.98	70.24	73.68	87.78	87.71	46.52	68.67	67.98	85.08
Flan-T5 + **TG**		78.47	70.13	65.77	70.03	69.98	75.18	88.73	88.72	47.05	68.75	68.05	85.25
Flan-T5 + **CLH**		77.16	73.67	62.15	64.36	62.21	63.94	89.19	89.19	45.88	67.24	67.63	84.96
Flan-T5 + **CLH** + **TG**		78.68	75.77	64.99	67.87	66.24	72.11	89.6	89.6	47.24	67.94	67.80	85.34
Flan-T5 + **CLH** + **TG** + **TF**		**81.53**†	**78.19**†	66.49	69.11	67.52	72.14	**90.09**†	**90.08**†	47.85	68.93	68.16	85.58
ToXCL		**81.53**†	**78.19**†	**68.11**†	**71.21**†	**70.27**†	**77.38**†	**90.09**†	**90.08**†	**49.03**†	**69.93**†	**68.85**†	**86.09**†

Table 3: Main experimental results. **CLH** stands for joint training with a classification head on top of the Flan-T5 Encoder. Our model, **ToXCL** is equivalent to Flan-T5 + **CL Head** + **TG** + **TF** + **CD** (all are described in Section 3.2). † denotes our model significantly outperforms *implicit toxic speech detection & explanation* baselines with p-value < 0.05 under t-test.

Model	Cor.↑	Flu.↑	Tox.↓
Flan-T5 (G2)	2.21	2.02	2.03
Flan-T5 (G3)	2.35	2.46	2.07
ToXCL	**2.56**	**2.63**	**1.97**
Kripp.'s alpha	0.81	0.84	0.78

Table 4: Human evaluation results.

present the cases discussed in Table 5. In cases (1), (3), (4), (7), (9), and (10) we observe that both GPT-2 and Flan-T5 fail to capture the intended meaning of the input posts, resulting in wrong detection or harmful explanations. In contrast, our **ToXCL** effectively captures the meaning of the posts, leading to accurate explanations that align closely with the ground truth ones. Notably, when comparing to the ground truth explanation, the explanation generated by **ToXCL** exhibits a more polite attitude (cases 2, 4, 8, 9) or even more accurate (case 3).

5.2 Ablation Studies

• **Target Group Generator (TG)** When adding the this module (i.e., + **TG** in Table 3), we observe that all baselines considered in our experiments achieve significant improvements compared to training them without the generated target groups. In addition, our Teacher Classifier also outperforms its variant trained on solely the input posts. These improvements emphasize the crucial role of the target group(s) in enhancing their ability to detect and explain implicit toxic speech. These results also consolidate our motivation for using generated target groups in guiding the models for solving the two tasks.

• **Teacher Forcing (TF)** By comparing the performance of Flan-T5 + **CLH** + **TG** before and after using *teacher forcing* (+ **TF** in Table 3), we observe that incorporating this technique improves the performance of the encoder of **ToXCL**. This improvement demonstrates that providing additional guidance to the encoder of **ToXCL** results in more accurate predictions and achieves a performance close to that of the *Teacher Classifier*.

• **Conditional Decoding Constraint (CD)** Finally, the impact of integrating the *Conditional Decoding Constraint* designed to avoid necessary explanation, is evaluated in Table 3. Compared with Flan-T5 + **CLH** + **TG** + **TF**, our **ToXCL** is improved significantly on the toxic explanation generation task. This confirms the effectiveness of *Conditional Decoding Constraint* in helping the outputs of our model to synchronize implicit toxic speech detection labels and toxic explanations.

5.3 Error Analysis

We present additional qualitative samples from both benchmarks in Table 5 to highlight key challenges faced by existing models in detecting implicit toxic speech and generating explanations. While our model performs well overall, there are still areas for improvement, as discussed below.

• **Coded Toxic Symbols** Our model, along with the baselines, struggles with detecting implicit toxic speech that contains abbreviated or coded to-

Index	Input Post	Ground Truth Explanation(s)	GPT-2 (IHC)	Flan-T5 (SBIC)	**ToXCL**
(1)	approx half muslms today result of inbreeding, perhaps this why they are ""hell bent"" on destroying christian based west? inbreed = low iq.	Muslims are result of inbreeding	Muslims are destructive		Muslims are inbred.
(2)	you hate like a jew. you are fixatd on hating white christians like a jew.	Jewish people have hate toward other groups	Jews hate white people		Jews don't like white poeple
(3)	funny how you never hear about white men raping women in black" neighborhoods. i know 2 white women who '	Blacks rape whites.	Blacks are inferior		White men are rapists.
(4)	she's another low iq hollywood liberal . islam is submission and btw if they ran america	Liberals are stupid.	None		Liberals are low IQ people.
(5)	rt. : " voters on obamas immigration plans : drop dead " " #noamnesty "	Immigrants ruin society	None		None
(6)	bruh i love white people too how do i join	Minorities are lesser races.	None		None
(7)	An Asian woman just gave me a handjob while fingering my ass Which isnt what i meant when i asked her to give me a 'Hand Roll and Pork Intestines. That's the problem with having the Chinese Massage place right next to the Chinese takeout place	['chinese women are sex workers.', "chinese women don't understand english."]		asians eat pork	asian women only care about sex
(8)	you see, i am a doctor. today i had to save a black guy's life i told his family: i did everything i could, but unfortunately he survived.	['black people are useless', 'black people need to be killed']		black people are worthless	black people aren't as good as whites
(9)	@BlakkHoodie_562 yeah that niggah played good, lowkey i wanted brazil to win	['black people are the n word']		None	niggah is a slang term
(10)	A bf and gf; are suppose to laugh; at these hoes; together;. A guy should never have these hoes; laughing; at his girl;	['degrades women by calling them names', 'women are called hoes', 'women are hoes']		None	women are hoes

Table 5: The qualitative explanation performance of GPT-2, Flan-T5, and **ToXCL** was evaluated on samples from the valid set of IHC and SBIC datasets.

kens, such as "*#noamnesty*" in case (5). This error has also been observed and discussed by previous work (ElSherief et al., 2021).

• **Misunderstood Sarcasm and Irony** Our model may face challenges in accurately detecting toxic sentences that contain indirect words. For example, case (6) involves the phrase *"bruh i love white people too how do i join"*, which uses indirect words such as "bruh" and "love" to express irony. The speaker sarcastically expresses a desire to join a racial group while implying that joining such a group is based on a belief in the superiority of white people or that minorities are lesser races.

• **Variant Explanations** In all cases except cases (5), (6), and (10), our **ToXCL** accurately identifies implicit toxic speech but generates linguistically different explanations from the ground truth(s). However, these generated explanations convey the same semantic meaning as the ground truth, indicating the model's ability to comprehend correctly the implicit meanings. This, along with discussions by Huang et al. (2022), demonstrates that instances of implicit toxic speech can have multiple correct explanations, highlighting the limitations of commonly-used n-gram evaluation metrics like BLEU-4 (Papineni et al., 2002) and ROUGE-L (Lin, 2004) scores.

6 Conclusion

We present **ToXCL**, a unified framework for implicit toxic speech detection and explanation (Sec. 3.2). It consists of three components: a Target Group Generator, an Encoder-Decoder model, and a Teacher Classifier. Our findings show that the Target Group Generator effectively identifies target groups, improving both accuracy and F1 scores for detecting implicit toxic speech. The novel encoder-decoder architecture successfully performs both tasks of detection and explanation without harming each other. The integration of the Teacher Classifier and the Conditional Decoding Constraint further enhances the performance of **ToXCL**, achieving state-of-the-art results in the task of toxic speech detection and explanation on two widely-used benchmarks. In the future, we will focus on addressing several limitations faced by our framework and baselines as specified in Section 5.3 to further enhance the performance of our model.

Limitations

Although our model has demonstrated strong performance, our error analysis (Sec. 5.3) has identified several challenges that still need to be addressed. One such challenge is the presence of coded toxic symbols, abbreviated words, or implicit phrases in the input posts, which may require external sources of knowledge for accurate interpretation. To address this, future work can focus on enhancing the models by incorporating additional reasoning capabilities and leveraging external knowledge. Additionally, existing evaluation metrics for the implicit toxic speech explanation task is also another limitation since this can be a one-to-many relationship problem in which there may have multiple non-overlapping but correct explanations for an implicit hate speech.

Ethical Considerations

While our method for implicit toxic speech has shown promise in identifying target groups, detecting implicit toxic speech, and generating explanations, it is crucial to consider the potential risks involved.

Firstly, there is a concern that the generated explanations may contain toxic words, depending on the training data (Sec. 5). This raises the possibility of the model spreading machine-generated toxic speech if it is misused. It is essential to address this toxicity to protect marginalized groups and shift power dynamics to the targets of oppression.

Secondly, there is a risk of reinforcing biases or amplifying harmful messages by providing explanations only for detected implicit toxic speech. If the model fails to detect implicit toxic speech, the absence of an explanation may imply acceptability or harmlessness. Considering explanations for all posts, regardless of detection, could be an approach to mitigate this risk, although our datasets do not provide explanations for non-toxic speech.

In conclusion, while having complete control over the **ToXCL**'s usage in real-world scenarios may not be feasible, it is essential to recognize and address potential risks. By doing so, our work offers an opportunity to combat harm directed at minority groups and empower targets of oppression. Therefore, addressing the quality of explanations in the training data is a critical step toward achieving our ultimate goal.

References

Badr AlKhamissi, Faisal Ladhak, Srini Iyer, Ves Stoyanov, Zornitsa Kozareva, Xian Li, Pascale Fung, Lambert Mathias, Asli Celikyilmaz, and Mona Diab. 2022. Token: Task decomposition and knowledge infusion for few-shot hate speech detection.

Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, Nicholas Joseph, Saurav Kadavath, Jackson Kernion, Tom Conerly, Sheer El-Showk, Nelson Elhage, Zac Hatfield-Dodds, Danny Hernandez, Tristan Hume, Scott Johnston, Shauna Kravec, Liane Lovitt, Neel Nanda, Catherine Olsson, Dario Amodei, Tom Brown, Jack Clark, Sam McCandlish, Chris Olah, Ben Mann, and Jared Kaplan. 2022. Training a helpful and harmless assistant with reinforcement learning from human feedback.

Esma Balkir, Isar Nejadgholi, Kathleen C Fraser, and Svetlana Kiritchenko. 2022. Necessity and sufficiency for explaining text classifiers: A case study in hate speech detection. *arXiv preprint arXiv:2205.03302*.

Satanjeev Banerjee and Alon Lavie. 2005. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pages 65–72, Ann Arbor, Michigan. Association for Computational Linguistics.

Valerio Basile, Cristina Bosco, Elisabetta Fersini, Debora Nozza, Viviana Patti, Francisco Manuel Rangel Pardo, Paolo Rosso, and Manuela Sanguinetti. 2019. SemEval-2019 task 5: Multilingual detection of hate speech against immigrants and women in Twitter. In *Proceedings of the 13th International Workshop on Semantic Evaluation*, pages 54–63, Minneapolis, Minnesota, USA. Association for Computational Linguistics.

Luke Breitfeller, Emily Ahn, David Jurgens, and Yulia Tsvetkov. 2019. Finding microaggressions in the wild: A case for locating elusive phenomena in social media posts. 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 1664–1674, Hong Kong, China. Association for Computational Linguistics.

Tommaso Caselli, Valerio Basile, Jelena Mitrović, and Michael Granitzer. 2021. HateBERT: Retraining BERT for abusive language detection in English. In *Proceedings of the 5th Workshop on Online Abuse and Harms (WOAH 2021)*, pages 17–25, Online. Association for Computational Linguistics.

Tommaso Caselli, Valerio Basile, Jelena Mitrović, Inga Kartoziya, and Michael Granitzer. 2020. I feel offended, don't be abusive! implicit/explicit messages

in offensive and abusive language. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 6193–6202, Marseille, France. European Language Resources Association.

Ying Chen, Yilu Zhou, Sencun Zhu, and Heng Xu. 2012. Detecting offensive language in social media to protect adolescent online safety. In *2012 International Conference on Privacy, Security, Risk and Trust and 2012 International Confernece on Social Computing*, pages 71–80.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Eric Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. 2022. Scaling instruction-finetuned language models. *arXiv preprint arXiv:2210.11416*.

Kevin Clark, Minh-Thang Luong, Urvashi Khandelwal, Christopher D. Manning, and Quoc V. Le. 2019. BAM! born-again multi-task networks for natural language understanding. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5931–5937, Florence, Italy. Association for Computational Linguistics.

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*.

Imre Csiszár. 1975. I-divergence geometry of probability distributions and minimization problems. *The annals of probability*, pages 146–158.

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.

Filip Karlo Dosilovic, Mario Bri, and Nikica Hlupic. 2018. Explainable artificial intelligence: A survey. *2018 41st International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, pages 0210–0215.

Mai ElSherief, Vivek Kulkarni, Dana Nguyen, William Yang Wang, and Elizabeth Belding. 2018. Hate lingo: A target-based linguistic analysis of hate speech in social media.

Mai ElSherief, Caleb Ziems, David Muchlinski, Vaishnavi Anupindi, Jordyn Seybolt, Munmun De Choudhury, and Diyi Yang. 2021. Latent hatred: A benchmark for understanding implicit hate speech. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 345–363, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Tracie Farrell, Miriam Fernandez, Jakub Novotny, and Harith Alani. 2019. Exploring misogyny across the manosphere in reddit. In *Proceedings of the 10th ACM Conference on Web Science*, pages 87–96.

Hao Fu, Shaojun Zhou, Qihong Yang, Junjie Tang, Guiquan Liu, Kaikui Liu, and Xiaolong Li. 2020. Lrc-bert: Latent-representation contrastive knowledge distillation for natural language understanding.

Njagi Dennis Gitari, Zhang Zuping, Hanyurwimfura Damien, and Jun Long. 2015. A lexicon-based approach for hate speech detection. *International Journal of Multimedia and Ubiquitous Engineering*, 10(4):215–230.

Ian Goodfellow, Yoshua Bengio, and Aaron Courville. 2016. *Deep learning*. MIT press.

Jianping Gou, Baosheng Yu, Stephen J. Maybank, and Dacheng Tao. 2021. Knowledge distillation: A survey. *International Journal of Computer Vision*, 129(6):1789–1819.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*.

Fan Huang, Haewoon Kwak, and Jisun An. 2022. Chain of explanation: New prompting method to generate higher quality natural language explanation for implicit hate speech. *arXiv preprint arXiv:2209.04889*.

Clayton Hutto and Eric Gilbert. 2014. Vader: A parsimonious rule-based model for sentiment analysis of social media text. In *Proceedings of the international AAAI conference on web and social media*, volume 8, pages 216–225.

Md Saroar Jahan and Mourad Oussalah. 2021. A systematic review of hate speech automatic detection using natural language processing. *arXiv preprint arXiv:2106.00742*.

Md Saroar Jahan and Mourad Oussalah. 2023. A systematic review of hate speech automatic detection using natural language processing. *Neurocomputing*, 546:126232.

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*, pages 5435–5442, Online. Association for Computational Linguistics.

Youngwook Kim, Shinwoo Park, and Yo-Sub Han. 2022. Generalizable implicit hate speech detection using contrastive learning. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 6667–6679, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

Klaus Krippendorff. 2011. Computing krippendorff's alpha-reliability.

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, Online. Association for Computational Linguistics.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. 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. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *International Conference on Learning Representations*.

Sean MacAvaney, Hao-Ren Yao, Eugene Yang, Katina Russell, Nazli Goharian, and Ophir Frieder. 2019. Hate speech detection: Challenges and solutions. *PLOS ONE*, 14(8):1–16.

Binny Mathew, Punyajoy Saha, Seid Muhie Yimam, Chris Biemann, Pawan Goyal, and Animesh Mukherjee. 2021. Hatexplain: A benchmark dataset for explainable hate speech detection. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(17):14867–14875.

J Nockleyby. 2000. 'hate speech in encyclopedia of the american constitution. *Electronic Journal of Academic and Special Librarianship*.

Alexandra Olteanu, Carlos Castillo, Jeremy Boy, and Kush Varshney. 2018. The effect of extremist violence on hateful speech online. In *Proceedings of the international AAAI conference on web and social media*, volume 12.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

Manoel Horta Ribeiro, Pedro H Calais, Yuri A Santos, Virgílio AF Almeida, and Wagner Meira Jr. 2018. Characterizing and detecting hateful users on twitter. In *Twelfth international AAAI conference on web and social media*.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. "why should i trust you?": Explaining the predictions of any classifier.

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*, pages 5477–5490, Online. Association for Computational Linguistics.

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.

Rohit Sridhar and Diyi Yang. 2022. Explaining toxic text via knowledge enhanced text generation. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 811–826, Seattle, United States. Association for Computational Linguistics.

Thanh Tran, Yifan Hu, Changwei Hu, Kevin Yen, Fei Tan, Kyumin Lee, and Se Rim Park. 2020. HABERTOR: An efficient and effective deep hatespeech detector. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7486–7502, Online. Association for Computational Linguistics.

Iulia Turc, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Well-read students learn better: On the importance of pre-training compact models.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

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, Florence, Italy. Association for Computational Linguistics.

Michael Wiegand, Josef Ruppenhofer, Anna Schmidt, and Clayton Greenberg. 2018. Inducing a lexicon of abusive words – a feature-based approach. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1046–1056, New Orleans, Louisiana. Association for Computational Linguistics.

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

Lijun Wu, Xu Tan, Di He, Fei Tian, Tao Qin, Jianhuang Lai, and Tie-Yan Liu. 2018. Beyond error propagation in neural machine translation: Characteristics of language also matter. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3602–3611, Brussels, Belgium. Association for Computational Linguistics.

Xinchen Yu, Eduardo Blanco, and Lingzi Hong. 2022. Hate speech and counter speech detection: Conversational context does matter. *arXiv preprint arXiv:2206.06423*.

Tianyi Zhang, Varsha Kishore, Felix Wu*, Kilian Q. Weinberger, and Yoav Artzi. 2020. Bertscore: Evaluating text generation with bert. In *International Conference on Learning Representations*.

A Data Pre-process

To facilitate our problem, we exclude instances of implicit toxic speech that lack an explanation, resulting in the removal of 844 samples from IHC and 8,220 samples from SBIC. Since the original IHC dataset does not include a designated test set, we create our own by randomly selecting 20% of the implicit toxic speech and non-toxic speech instances. The final statistics of both datasets are shown in Table 6.

Split	# toxic	# non-toxic	# samples
IHC Train	5,002	10,633	15,635
IHC Valid	1,254	2,658	3,912
SBIC Train	12,098	16,698	28,796
SBIC Dev	1,806	2,054	3,860
SBIC Test	1,924	1,981	3,905

Table 6: Statistics of `Implicit Hate Corpus` and `Social Bias Inference Corpus` after pre-processing.

B Evaluation Algorithm

We present our evaluation metrics to evaluate the explanation generation capability of the models. To penalize unnecessary explanations for non-toxic speech, we add 100 to every score when the label is "None" and the model generation output is "None".

Algorithm 1: Computations of Explanation Generation Metrics

```
Input: b_gen_labels, b_gen_expls
Initialize:
N = len(b_gen_labels)
s_bleu = s_rouge = s_meteor = s_bertscore = 0
for idx in range(N) do
    label = b_gen_labels[idx]
    expl = b_gen_expls[idx]
    if label == "None" and expl == "None" then
        add 100 to s_bleu, s_rouge, s_meteor, s_bertscore
    end
    else if label ≠ "None" and expl ≠ "None" then
        s_bleu += bleu(label, expl)
        s_rouge += rouge(label, expl)
        s_meteor += meteor(label, expl)
        s_bertscore += bertscore(label, expl)
    end
end
return s_bleu/N, s_rouge/N, s_meteor/N, s_bertscore/N
```