# Mitigating Text Toxicity with Counterfactual Generation

**Anonymous ACL submission**


## Abstract

Toxicity mitigation consists in rephrasing text in order to remove offensive or harmful meaning. Neural natural language processing (NLP) models have been widely used to target and mitigate textual toxicity. However, existing methods fail to detoxify text while preserving the initial non-toxic meaning at the same time. In this work, we propose to apply eXplainable AI (XAI) methods to both target and mitigate textual toxicity. We propose `CF-Detox`$_{\text{tigtec}}$ to perform text detoxification by applying local feature importance, counterfactual example generation and counterfactual feature importance methods to a toxicity classifier distinguishing between toxic and non-toxic texts. We carry out text detoxification through counterfactual generation on three datasets and compare our approach to three competitors. Automatic and human evaluations show that recently developed NLP counterfactual generators lead to competitive results in toxicity mitigation. This work is the first to bridge the gap between counterfactual generation and text detoxification and paves the way towards more practical applications of XAI methods.


## 1 Introduction

Online textual toxicity can be considered as rude, aggressive and degrading attitudes exhibited on online platforms, ranging from harmful to hateful speech. Hateful speech is defined as aggressive or offensive language against a specific group of people who share common characteristics, such as religion, race, gender, sexual orientation, sex or political affiliation (Castaño-Pulgarín et al., 2021). Such toxic content has proliferated on the Internet in the recent years (Thomas et al., 2021), raising concerns about its multi-faceted negative impact, such as the potential to threaten the psychological and physical well-being of victims (Walther, 2022) or to be used as a medium for criminal actions (Rapp, 2021).

Figure 1: Example of one text detoxification through counterfactual generation with our proposed `CF-Detox`$_{\text{tigtec}}$ compared to `MaRCo` (Hallinan et al., 2023) and `CondBERT` (Dale et al., 2021). Text changes to mitigate toxicity are highlighted in blue. Explicitly toxic words have been censored with *.

To cope with the rapid development of online toxic content and curb its societal impact, automatic toxicity processing methods have been developed to detect and process hateful content in online communities and digital media platforms (Fortuna and Nunes, 2018). For example, such tools can be used to clean up language model training datasets, or to suggest interventions on toxic and hateful messages to the human-labor moderating online platforms. In particular, text detoxification (or toxicity mitigation) aims to rewrite toxic text in order to remove (or to mitigate) toxicity while preserving the initial non-toxic meaning and maintaining grammatical plausibility. Recent methods based on neural NLP models have been developed to perform text detoxification (Hallinan et al., 2023; Dale et al., 2021; Laugier et al., 2021) by either generating text under constraint, or by detecting toxic content

and modifying it. While these methods succeed in drastically lowering textual toxicity, they generally fail to preserve the initial non-toxic content.

In this paper we propose to address toxicity targeting and mitigation by applying eXplainable AI (XAI) methods (Molnar, 2020): more precisely Local Feature Importance (LFI) (Barredo Arrieta et al., 2020), counterfactual (CF) generators (see Guidotti (2022) for a survey) and Counterfactual Feature Importance (CFI) (Bhan et al., 2023b). The former aims at detecting important input features to explain a model prediction. Instead, CF example generation explains a model prediction by identifying the minimal changes that enable flipping the outcome of a classifier. CFI measures token modification importance to explain label flipping related to CF explanations.

The main contributions of this work are as follows: (*i*) we show that LFI and CF generation can be applied to a toxicity classifier to target toxicity and to perform toxicity mitigation respectively, (*ii*) we propose `CF-Detox`$_{\text{tigtec}}$, a detoxification method based on a recently developed CF generator: `TIGTEC` (Bhan et al., 2023c) and validated by both automatic and human experiments, (*iii*) we illustrate how to apply CFI methods to make detoxified texts even more content preserving.

For illustrative purposes, Figure 1 shows an initial toxic text and its detoxified versions obtained from our proposed method, `MaRCo` (Hallinan et al., 2023) and `CondBERT` (Dale et al., 2021) respectively. Here, counterfactual detoxification leads to more plausible, sparse and context preserving text as compared to the other methods.

The paper is organized as follows, in Section 2 we first recall basic principles about automatic toxicity processing and XAI. In Section 3 we propose methods to use common XAI approaches to (1) target toxicity and (2) generate plausible and content-preserving detoxified texts. Experimental results discussed in Section 4 highlight that text detoxification through CF generation achieves competitive results in terms of toxicity mitigation, content preservation and plausibility, as compared to state-of-the-art competitors. We show that CFI can generate more context preserving detoxified texts.

## 2 Background and Related Work

This section recalls the task of automatic toxicity detection and mitigation. We presents existing methods aiming to detoxify text using neural NLP models. We then the XAI principles used in the next section to perform toxicity detection and mitigation. In the following, we employ *text detoxification* and *toxicity mitigation* interchangeably.

### 2.1 Automatic Toxicity Processing

**Definition and objective** Textual toxicity can be defined in multiple ways (Fortuna and Nunes, 2018) and can take various forms, such as rude, offensive or hateful speech, potentially causing online harm to isolated people or minority groups (Thomas et al., 2021). Automatic toxicity processing can essentially take two forms: detection and mitigation. **Toxicity detection** can be either based on prior knowledge (vocabulary, regex) or obtained from a fine tuned toxicity classifier $f : \mathcal{X} \rightarrow \mathcal{Y}$ mapping an input text representation space $\mathcal{X}$ to an output space $\mathcal{Y}$ to distinguish toxic and non-toxic texts (Cook et al., 2023). Training language models to perform this classification is difficult, as it requires access to datasets labeled based on an idiosyncratic definition of toxicity derived from human annotators (Jigsaw, 2018). **Toxicity mitigation** consists in rewriting a toxic text while preserving the non-toxic meaning. This task is even more difficult because it requires disentangling toxic and non-toxic meanings to modify the former plausibly while preserving the latter.

**Evaluation criteria of text detoxification** Numerous desirable properties have been proposed to assess automatically text detoxification. We organize them into three categories. When ground truth detoxified texts are unavailable, we use the previously introduced $f$ classifier as an oracle to evaluate the toxicity level of the supposedly detoxified text. *Accuracy* (**ACC**) measures the extent to which the generated texts are accurately detoxified with respect to $f$. *Accuracy* can be measured by computing either the rate of successful changes or the average toxicity score of the generated texts using $f$. *Proximity* or *content preservation* (**CP**) evaluates how close two texts are. Textual similarity can be defined in two ways. The first one consists in evaluating textual proximity based on word sequence co-occurrences, with metrics such as self-BLEU (Papineni et al.,

2002) or the Levenshtein distance. The second consists in measuring semantic similarity from word-level embeddings (Pennington et al., 2014) or sentence-level embeddings (Zhang et al., 2019). Finally, text *plausibility* is automatically evaluated by the perplexity (**PPL**) score usually obtained from generative language models such as GPT-2 (Radford et al., 2018).

### 2.2 Toxicity Mitigation with Neural NLP Models

This section describes two categories of methods leveraging neural NLP models to perform detoxification, namely Text Style Transfer (TST) and Masking and Reconstructing (M&R).

**Text Style Transfer** TST (see (Hu et al., 2022) for a survey) aims to alter the stylistic attributes of an initial text while preserving its content that is unrelated to the target style. Text detoxification can be achieved through TST, where the initial style is characterized by the presence of toxicity, and the target style is defined by its absence. TST is usually performed by generating text with neural NLP decoders guided by an NLP toxicity classifier. In general, TST methods vary based on the language model used for autoregressive text generation and the heuristic of text generation steering.

A first TST approach (Nogueira dos Santos et al., 2018) uses an encoder-decoder architecture based on recurrent neural networks to generate non-toxic text, using a toxicity convolutional neural network classifier to steer the style transfer. Another method (Laugier et al., 2021) fine-tunes a text-to-text T5 model (Raffel et al., 2019) by using a denoising and cyclic auto-encoder loss. `ParaGeDi` (Dale et al., 2021) uses a pre-trained T5-based paraphraser model and a class-conditioned language model to steer the text generation. TST methods generally detoxify text accurately but struggle to preserve its non-toxic meaning (Hallinan et al., 2023).

**Masking and Reconstructing** Masking and Reconstructing (M&R) approaches perform toxicity mitigation by sequentially (1) targeting toxic content, (2) masking it, and (3) modifying it. Once the toxic content is targeted, mask infilling is usually performed with a neural NLP encoder. In general, M&R methods differ in the way they target toxic content, and the neural NLP model used to perform mask infilling.

A first M&R approach (Tran et al., 2020) detoxifies text by retrieving potential harmful Part-Of-Speech (POS) based on a predefined toxicity vocabulary, generating non-offensive POS substitution candidates, and editing the initial text through mask infilling with a RoBERTa encoder for unacceptable candidates. `CondBERT` (Dale et al., 2021) identifies tokens to be masked using a logistic bag-of-words classifier and performs mask infilling using a BERT encoder. The most recent M&R method called `MaRCo` (Hallinan et al., 2023) detects POS that could convey toxic meaning by comparing likelihoods from two BART encoder-decoders respectively fine-tuned on toxic and non-toxic content. The targeted potential toxic content is then replaced by non-toxic content by mixing token probabilities from these two encoder-decoders and a third neutral model. On average, M&R yields to better results than TST in terms of content preservation (Hallinan et al., 2023).

### 2.3 XAI for NLP

In the following, we consider the neural NLP toxicity classifier $f : \mathcal{X} \rightarrow \mathcal{Y}$ introduced in the previous section, and a text $x = [t_1, ..., t_d] \in \mathcal{X}$ represented as a sequence of tokens with $f(x) = y$. $\mathcal{Y}$ can either be a binary space that distinguishes toxic and non-toxic texts or a multi-class space categorizing several levels of toxicity.

**Local Feature Importance** A Local Feature Importance (LFI) function $g : \mathcal{X} \rightarrow \mathbb{R}^d$ explains a prediction by a vector $[z_1, ..., z_d]$ where $z_i$ is the contribution of the $i-$th token to the prediction. The higher the contribution, the more important the token is to explain the prediction of the classifier $f$. Three types of LFI methods can be distinguished: *perturbation-based* such as KernelSHAP (Lundberg and Lee, 2017), *gradient-based* such as Integrated gradients (Sundararajan et al., 2017) and *attention-based* such as self-attention (Bhan et al., 2023a) if $f$ is a Transformer.

**Counterfactual explanations** Counterfactual (CF) explanations emphasize what should be different in an input instance to change the associated outcome of a classifier (Ali et al., 2023). CF examples provide contrastive explanations by simulating alternative instances to assess whether a specific event (in our case the predicted class) still occurs or not (Miller, 2019). For a given classifier $f$ and an instance of interest $x$, a CF example $x_{\text{cf}}$

must be close to $x$ but predicted differently. It is defined as:

$$x_{\text{cf}} = \underset{z \in \mathcal{X}}{\text{argmin}}\, c(x, z) \;\text{ s.t. }\; f(z) \neq f(x) \qquad (1)$$

where $c : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$ is a cost function that aggregates several expected CF characteristics, such as the textual distance. The CF explanation is then the difference between the generated CF example and the initial data point, $x_{\text{cf}} - x$. Many desirable characteristics for CF explanations have been proposed (Guidotti, 2022), such as sparsity or plausibility to make sure that the CF example is not out-of-distribution (Laugel et al., 2019). In Section 3, we present several methods to generate textual CFs by comparing and relating them with the toxicity mitigation task.

**Counterfactual Feature Importance** A Counterfactual Feature Importance (CFI) operator $h : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}^d$ computes token substitution importance between an instance of interest and its related CF explanation (Bhan et al., 2023b). In the same way as common LFI, $h_i$ is the contribution of the $i$-th token modification. CFI highlights the most important token modifications, making non-sparse CF explanations easier to read and more intelligible. A simple but effective way to compute CFI is to apply Integrated gradients to explain the difference between $x_{cf}$ and $x$, by setting the baseline to the initial instance $x$.

## 3 When XAI Meets Text Detoxification

In this section we show how LFI methods can foster toxic content targeting and discuss how to apply CF generation methods for performing text detoxification. We also propose a new method based on CFI to be more content preserving.

### 3.1 Targeting Toxic POS with LFI

Toxic POS targeting consists in identifying the elements in a toxic text that induce its toxicity. While toxicity can be easily defined partially by a predefined lexical field, it can also take on more complex forms implying specific cultural references or sarcasm that are difficult to detect automatically. Let $f$ be a toxicity classifier and $x$ is a toxic text with $f(x) = \texttt{toxic}$. We propose to apply LFI to $f$ to highlight important tokens that explain why $x$ has been classified as toxic by $f$, enabling toxic POS detection. This way, toxic POS detection with LFI does not require the definition

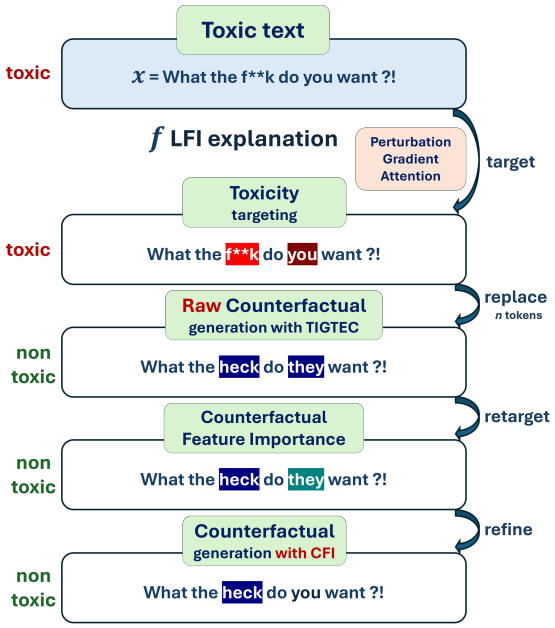


Figure 2: Illustrative example of a toxic text and a CF detoxification process with our proposed *target-then-replace-then-refine* approach. (1) Toxic content (in red) is targeted and (2) then modified (in blue). To be more content preserving, the counterfactual example is finally refined (3) by restarting the counterfactual generation process, guided by CFI. Explicitly toxic words have been censored with *. The darker the shade of color (red for LFI, blue for CFI), the higher the importance.

of a predefined toxicity vocabulary and is only based on a model that is trained to discriminate between toxic and non-toxic texts. Then, toxicity is detected in a *data-driven* manner based on a fine-tuned neural NLP model.

The top part of Figure 2 (toxicity targeting) shows an example of toxic POS targeting with LFI applied to a toxicity classifier: the token "**f**k**" is assessed as important to predict toxicity.

### 3.2 Epistemic Similarities Between Text Detoxification and CF Generation

In the following, we show how toxicity mitigation and CF generation share common characteristics.

**Similarity of objectives and evaluation criteria.** Firstly, the aim of both toxicity mitigation and CF generation is to generate text close to an initial instance, while reaching a target state. For detoxification purpose, the target state is defined as non-toxicity, whereas the target state is defined by the $f$ classifier label for CF generation. Secondly, as discussed in Section 2, toxicity mitigation methods share with CF generators

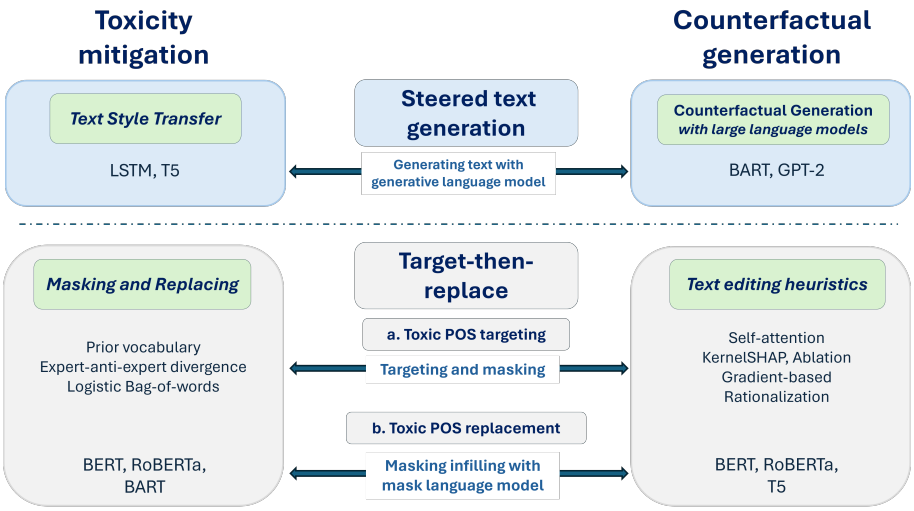


Figure 3: Toxicity mitigation and counterfactual generation comparison by method category. Toxicity mitigation methods and counterfactual generators can be categorized as *steered text generation* and *target-then-replace* approaches. Neural NLP models used to generate text or replace tokens are similar or of the same nature.

several evaluation criteria, such as *accuracy*, *proximity* and *plausibility*. These evaluation criteria are measured with the same metrics, such as sparsity, embedding-based semantic similarity, BLEU score or perplexity measures.

**Similarity of constitutive categories of methods.** Thirdly, methods detoxifying text and generating CFs can be grouped into two similar categories. As mentioned in (Bhan et al., 2023c), textual CF generators can be of two types: *Text editing heuristics* and *Counterfactual generation with large language models*.

*Text editing heuristics* is a family of CF generation methods that builds CF examples by slightly modifying the input text whose prediction is to be explained. Important tokens are targeted and modified with mask language models to switch the outcome of the classifier, making this approach very similar to the M&R way of detoxifying text. Text editing heuristics methods differ in the way they target important tokens and the language model used to modify the initial text. Regarding the former methods, they mostly target tokens to be modified by applying LFI methods to the classifier that has to be explained. For example, `CLOSS` (Fern and Pope, 2021) applies Ablation, `MiCE` (Ross et al., 2021) *gradient-based* approaches, `CREST` (Treviso et al., 2023) leverages rationalization methods, and `TIGTEC` (Bhan et al., 2023c) employs KernelSHAP or Self-attention. Next, mask language models used to perform mask infilling are essentially T5, RoBERTa or BERT. This way, M&R methods and Text editing heuristics differ only in the way they target toxic POS. As mentioned in the previous section, the former detect tokens to change either by the use of prior knowledge (vocabulary), intrinsically interpretable model (logistic model) or expert-anti-expert disagreement, whereas Text editing heuristics apply LFI methods to a toxicity classifier. Finally, text is modified using the same kind of NLP encoder. We propose to group these two kinds of approaches together under the name of "**target-then-replace**" methods. Figure 2 shows the raw *target-then-replace* CF detoxification process, where the detoxification consists in performing the two following token changes: **f**k → heck** and **you → they**, leading to a text certainly detoxified, but not perfectly content preserving.

*Counterfactual generation with large language models* (CF-LLM) approaches build CF examples by leveraging pre-trained generative language models in the same way as TST text detoxifiers. These methods differ in the language model applied to generate text and the heuristic used to steer the model towards a specific objective. For example, `CASPer` (Madaan et al., 2022) learns perturbations to steer text generation with BART (Lewis et al., 2020) and `Polyjuice` fine-tunes GPT-2 to generate CF examples. Thus, TST and CF-LLM methods differ in the way generative language models are steered towards a specific *style* or *label*. We group these two families of approaches together under the name of "**steered text generation**" methods.

Figure 3 summarizes the connection between TST text detoxification and CF-LLM generation on one hand, and M&R methods and Text editing heuristics on the other.

All in all, CF generation and text detoxification can be (1) defined in the same way with the objective to find a small change to reach a target state, (2) evaluated with common metrics and (3) categorized in two similar families of methods, namely *steered text generation* and *target-then-replace*.

### 3.3 Counterfactual Toxicity Mitigation Methodology

We propose to perform toxicity mitigation with a *target-then-replace* CF generator with respect to the $f$ toxicity classifier. We postulate that, following the notations of Equation 1, toxicity mitigation can be performed by setting $x$ as a toxic instance of interest with $f(x) = \texttt{toxic}$. Therefore, the objective is to find $x_{\text{cf}}$ that minimizes the cost function $c$ such that $f(x_{\text{cf}}) = \texttt{non-toxic}$. This way, by denoting $\mathcal{M}$ as a *target-then-replace* CF generation method and $g$ the LFI method used to target important tokens to be modified, $x_{cf}$ is defined such as $x_{cf} = \mathcal{M}(x, h(x))$ . The lack of ground truth detoxified texts is then overcome through the use of $f$ as an oracle to guide text detoxification. This way, text detoxification through CF generation consists in detecting texts classified as toxic by $f$ and generating their related detoxified CF examples.

We also propose to use CFI to refine CF explanations (detoxified texts) and make them sparser, i.e. more content preserving. Since CF generation methods can generate explanations involving unnecessary token modifications (see token "**you**" in Figure 2), we re-run the CF generator, guided by CFI to optimally target the tokens that have to be modified to perform detoxification. Formally, refining a CF example with a CFI operator consists in generating $\tilde{x}_{cf}$ such as $\tilde{x}_{cf} = \mathcal{M}(x, h(x_{cf}, x))$ with $h$ a CFI operator. We categorize this methodology as a *target-then-replace-then-refine* approach. The entire Figure 2 traces the whole *target-then-replace-then-refine* process of CF detoxification.

## 4 Experimental Settings

We compare our approach performing CF toxicity mitigation named $\texttt{CF-Detox}_{\text{tigtec}}$ to three competitors: `MaRCo`, `CondBERT` and `ParaGeDi`.

### 4.1 Experimental Protocol

**Datasets** We perform toxicity mitigation on three toxicity datasets from Hallinan et al. (2023). **Microagression.com** (MAgr) is a public blog containing socially-biased interactions with offending quotes. **Social Bias Frames** (SBF) is a corpus of offensive content from various online sources. We use a subset of SBF where the texts have been labeled as hateful by annotators. **DynaHate** is a dataset of hate comments that are difficult to detect for a hate-speech classifier. Toxicity mitigation is run on texts initially classified as toxic in all three cases.

**Counterfactual generator and competitors** We instantiate our method proposed in the previous section by setting as CF generation method `TIGTEC`, since it achieves the best compromise in terms of accuracy, content preservation and plausibility when compared with other major competitors (Bhan et al., 2023c). `TIGTEC` is a *target-then-replace* textual CF generator that implements several LFI methods to target important tokens to be changed and explore the solution space with a tree search policy. We run two versions of `TIGTEC` based on either a BERT or a DistilBERT trained on a toxic task dataset from Kaggle[1] to learn to distinguish toxic texts from non-toxic ones.

Toxicity mitigation is then performed by generating CF examples starting from toxic texts in order to reach a non-toxic state. In the following, we call $\textbf{CF-Detox}_{\textbf{tigtec}}$ our new toxicity mitigation method based on CF generation. For each neural model, $\textbf{CF-Detox}_{\textbf{tigtec}}$ is run by using KernelSHAP to target toxic POS. We show the results obtained with other LFI methods in Appendix A.3. We compute CFI to refine the raw CF examples generated by $\textbf{CF-Detox}_{\textbf{tigtec}}$ by explaining the differences between the initial instances and their raw detoxified versions via Integrated gradients. CFI is only applied to CF examples (raw detoxified texts) initially involving at least two token modifications.

$\textbf{CF-Detox}_{\textbf{tigtec}}$ is compared to three state-of-the art text toxicity mitigation methods: **`MaRCo`** (Hallinan et al., 2023), **`CondBERT`** and **`ParaGeDi`** (Dale et al., 2021).

[1] https://www.kaggle.com/datasets/rounak02/imported-data

| Dataset | Metric | CondBert | MaRCo | ParaGeDi | CF-Detox$_{tigtec}$ | |
|---|---|---|---|---|---|---|
| | | | | | Bert-kshap | Distilbert-kshap |
| Dynahate | ΔPPL ↓ | 1.37 | 0.70 | **0.43** | 1.24 | 1.25 |
| | %CP ↑ | 68.8 | 68.4 | 71.2 | 86.4 | **86.6** |
| | %S ↑ | 64.1 | 67.8 | 31.1 | 86.1 | **86.3** |
| | %ACC ↑ | **92.3** | 70.2 | 89.9 | 82.9 | 81.9 |
| | SCORE ↓ | **0.10** | 0.30 | 0.14 | 0.21 | 0.23 |
| MAgr | ΔPPL ↓ | 2.59 | 0.72 | **0.69** | 2.04 | 2.16 |
| | %CP ↑ | 66.2 | 65.2 | 78.4 | 84.6 | **86.1** |
| | %S ↑ | 47.3 | 66.0 | 33.0 | **70.1** | **70.1** |
| | %ACC ↑ | **98.1** | 82.1 | 91.6 | 95.1 | 91.8 |
| | SCORE ↓ | **0.05** | 0.20 | 0.13 | 0.10 | 0.14 |
| SBF | ΔPPL ↓ | 1.48 | 0.82 | **0.59** | 1.27 | 1.26 |
| | %CP ↑ | 75.6 | 74.7 | 75.6 | 90.2 | **90.4** |
| | %S ↑ | 68.1 | 68.1 | 37.3 | 87.6 | **88.4** |
| | %ACC ↑ | **95.9** | 70.2 | 91.7 | 87.6 | 84.2 |
| | SCORE ↓ | **0.07** | 0.31 | 0.12 | 0.16 | 0.21 |

Table 1: Counterfactual toxicity mitigation comparison to competitors on three datasets.

**Automatic evaluation** We use the 5 metrics previously introduced to assess toxicity mitigation. The toxicity metrics are based on a pre-trained toxicity classifier. This toxicity classifier is different from the one used to steer CF-Detox$_{tigtec}$ toxicity mitigation. More information about the toxicity classifier used can be found in Appendix A.2. The success rate is computed with the accuracy (**%ACC**) from the binary classifier to assess if the evaluated text is toxic or non-toxic. **%ACC** is defined as the the percentage of non-toxic texts, with respect to the pre-trained toxicity classifier. The average toxicity score (**SCORE**) is obtained from the classification layer.

The sparsity (**%S**) is computed with the word-based Levenshtein distance. The content preservation (**%CP**) is computed with the cosine similarity between Sentence Transformer (Reimers and Gurevych, 2019) embeddings to evaluate the semantic proximity between the initial toxic text and its detoxified version. Text plausibility is measured with the perplexity score (Jelinek et al., 2005) and compared to the perplexity of the original text (**ΔPPL**). This way, a ΔPPL score lower than 1 indicates than the text plausibility increases whereas ΔPPL greater than 1 means that the detoxified text is less plausible.

**Human-grounded evaluation** In addition to automatic evaluation, we perform a human-grounded experiment to compare CF-Detox$_{tigtec}$ to MaRCo, CondBERT and ParaGeDi in terms of toxicity mitigation performance. It consists in asking 5 annotators to rank detoxified texts by toxicity level obtained by applying CF-Detox$_{tigtec}$, MaRCo, CondBERT and ParaGeDi on 20 randomly selected texts from each dataset. The order of appearance of the toxicity mitigation methods and the dataset is randomized, so that there is no spatial bias in information processing. Annotators can rank texts at the same level if necessary. We give more information about the human-grounded experiment in Appendix A.4.

### 4.2 Results

**Global results** Table 1 shows the experimental results obtained by running each method on the same datasets. In this table, CF-Detox$_{tigtec}$ has been run using KernelSHAP to target toxic POS. For each dataset, CF-Detox$_{tigtec}$ leads to the most content preserving texts, with the highest %CP and %S scores. On the other hand, CF-Detox$_{tigtec}$ performs in average worse than CondBERT in terms of detoxification accuracy and score across all datasets. Still, the toxicity of texts generated by CF-Detox$_{tigtec}$ is in average lower than that of MarCo and similar to ParaGeDi over all text corpora. If CondBERT mitigates the most toxicity, the resulting detoxified texts are significantly different from the initial ones in terms of sparsity and semantic proximity. CondBERT generates the less plausible text across all datasets and degrades text plausibility whereas ParaGedi and MaRCo improve it. In particular, ParaGeDi produces the most plausible detoxified text. This result stems from the autoregressive properties of the paraphrase language model used by ParaGeDi to generate text that is intended to be plausible. However, the text generated by ParaGeDi diverges

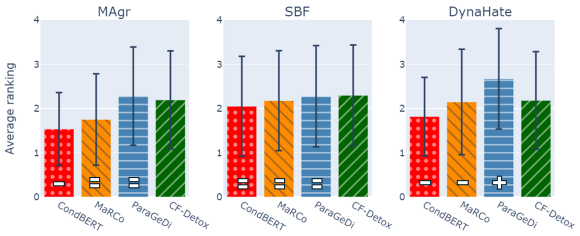


Figure 4: Counterfactual toxicity comparison with a human-grounded experimental ranking evaluation. Competitor rank distributions are compared to CF-Detox$_{\text{tigtec}}$ using a one-tailed paired t-test with a 5% threshold risk. The sign "-" indicates that the rank is lower in average as compared to CF-Detox$_{\text{tigtec}}$, whereas "=" and "+" respectively indicate that the ranking is in average similar and greater.

the most from the initial text.

The high perplexity level of the text generated by CF-Detox$_{\text{tigtec}}$ can be partially attributed to the mask language model used for generating new text. Indeed, it is significantly smaller than the encoder-decoder models used by ParaGeDi and MaRCo for text generation. The model used by CF-Detox$_{\text{tigtec}}$ is a *small* 66M parameters DistilBERT for masked language model, whereas each encoder-decoder model used by MaRCo and ParaGeDi to generate text are respectively a 139M parameters BART and a 220M parameters T5. Using a bigger mask language model such as BERT-base or BERT-large would improve the plausibility of the text generated by CF-Detox$_{\text{tigtec}}$. CF-Detox$_{\text{tigtec}}$ still generates significantly more plausible texts as compared to CondBERT.

**Human evaluation** Figure 4 shows the results from the human-grounded experiment where human annotators rank methods' outputs by level of toxicity. CondBERT achieves the lowest level of toxicity on DynaHate and MAgr, which is consistent with the automatic analysis. CF-Detox$_{\text{tigtec}}$ and MaRCo produce less toxic texts as compared to ParaGeDi on the DynaHate dataset. Toxicity is overall at the same level across CF-Detox$_{\text{tigtec}}$, MaRCo and ParaGeDi on SBF.

This way, automatic and human evaluation underpin the fact that CF-Detox$_{\text{tigtec}}$ offers another possible compromise between toxicity, context preservation and text plausibility as compared to other state-of-the-art existing methods.

**CFI enhancement** We refine detoxified texts using CFI following the *target-then-replace-then-refine* methodology previously introduced. Table 2 shows similarity and sparsity growth rate obtained from CFs obtained by running CF-Detox$_{\text{tigtec}}$ with KernelSHAP to target toxic POS. On average, CFI increases both similarity and sparsity across all datasets, when CF-Detox$_{\text{tigtec}}$ is either based on BERT or DistilBERT. However, the gains are higher with the DynaHate dataset (respectively +4.7% and +4.8% uplift in sparsity for BERT and DistilBERT) containing the most toxic texts as compared to MAgr and SBF. We give more information about the impact of CFI enhancement on CF-Detox$_{\text{tigtec}}$ performance on Appendix A.5.

| Model | Dataset | Similarity | Sparsity |
|---|---|---|---|
| BERT | MAgr | +0.5% | +1.2% |
| | SBF | +0.4% | +2.2% |
| | DynaHate | **+1.5%** | **+4.7%** |
| DistilBERT | MAgr | +0.3% | +0.9% |
| | SBF | +0.5% | +2.2% |
| | DynaHate | **+2.0%** | **+4.8%** |

Table 2: CF-Detox$_{\text{tigtec}}$ sparsity and similarity improvement with CFI refining.

**Ablation study** We evaluate the sensitivity of CF-Detox$_{\text{tigtec}}$ performance to the LFI method used to target toxic content in the initial text. It turns out that Self-attention, Integrated-gradients and KernelSHAP all lead to similar results in terms of toxicity mitigation, content preservation and text plausibility. These results underpin CF-Detox$_{\text{tigtec}}$ robustness. These results are analyzed in greater depth in Appendix A.3 based on Table 3.

## 5 Conclusion

This paper formalized how LFI and CF generation methods and CFI can be leveraged to accurately target textual toxic content and perform toxicity mitigation. CF-Detox$_{\text{tigtec}}$ leads to competitive results, with state-of-the-art performance in terms of content preservation while accurately detoxifying text and generating plausible text. CF-Detox$_{\text{tigtec}}$ is versatile since it can be used with various types of LFI methods (such as attention, gradient and perturbation) to target toxic POS and any kind of classifier allowing LFI computation.

This paper is the first to show the extent to which fields such as automatic toxicity processing and explainable AI, which have developed in parallel, actually share epistemic similarities and can be mutually beneficial, paving the way towards practical applications of XAI methods.

## 6 Limitations

**Classifier performance dependence** CF toxicity mitigation is highly dependent on the $f$ toxicity classifier used to steer detoxification. If $f$ is unreliable and only performs well on its training set, the risk of incorrectly indicating that the text has been detoxified is high. Therefore, the choice of $f$ and the toxicity training data must be made with caution to avoid incorrect toxicity assessments during detoxification.

**Use of other CF generators** CF detoxification has been tested by applying `TIGTEC`. Other CF generators such as `MiCE` (Ross et al., 2021), `CREST` (Treviso et al., 2023) or `CLOSS` (Fern and Pope, 2021) could be used to perform toxicity mitigation. We believe that these CF generation methods could lead to other levels of compromise between toxicity lowering, text plausibility and content preservation.

**Acting on the effects, not the causes** Online toxicity is a systemic problem with complex and multiple roots (Salminen et al., 2018). Automatic toxicity processing does not in itself solve the factors *causing* online toxic content generation, but they offer technological means of *adaptation* to its rapid development. While CF toxicity mitigation may yield competitive results, it also raises ethical concerns that we discuss in Section 7.

## 7 Ethical Considerations

**Managing diversity of values** The perceived toxicity of language can vary based on identity and beliefs (Al Kuwatly et al., 2020). For instance, conservative annotators can show a higher propensity to label African American English dialect as toxic while being less likely to annotate anti-Black comments as harmful (Sap et al., 2022). Such annotator biases can be reflected in the datasets used to detect toxicity. Toxicity training sets have to be built by annotators carefully selected to represent diversity of values. Besides, toxicity training sets have to be chosen in order to to make sure that the values implicitly encoded in the classifier match those expected.

**Against malicious use** A toxicity mitigation tool can be misused in various ways. One example is to use it as an adversarial attack generator to make a toxic text seem non-toxic. Adversarial attacks are small perturbations of data instances fooling a classifier with imperceptible changes (Liang et al., 2022), which bring them formally close to CFs. Since toxicity mitigation methods are based on the use of a toxicity classifier to steer the detoxification process, these methods are subject to adversarial attacks. In this manner, a dishonest user could hijack a detoxification tool to find small modifications to the initial texts, leading a toxicity classifier to falsely assess that a text is correctly detoxified. A recent work (Colombo et al., 2023) proposes a method to detect textual adversarial attacks.

**Favoring human-in-the-loop to overcome detoxification inaccuracy** State-of-the-art toxicity mitigation algorithms do not remove toxicity perfectly. Therefore, deploying a toxicity mitigation algorithm fully automatically is risky since it could let harmful content spread. These tools should be used as the first layer of hate content processing before integrating humans into the loop. Behind any platform, moderation is partly performed by human labor (Gillespie, 2018; Jhaver et al., 2023). By being exposed to disturbing toxic content, human moderators can develop psychological and emotional distress (Spence et al., 2023). Toxicity mitigation tools have the potential to induce a socio-technical change, suggesting textual intervention and reducing exposure for content moderators. We suggest using hate content detectors and mitigation methods based on the level of toxicity of the text. Text with the highest level of toxicity could simply be deleted, as it would be unlikely to be modified without completely altering its original meaning, whereas intermediate toxicity levels could be handled by a toxicity mitigation algorithm to suggest a softened version. This way, content moderators' exposure to the most hateful content would be significantly limited, and the more ambiguous content would be preprocessed by the mitigation algorithm to propose a more acceptable first version.

**Personal data** Each voluntary participant to the human evaluation signed an informed consent form outlining the project' details. The data were anonymized and processed only by the authors and stored in accordance with the General Data Protection Regulation (GDPR). It was possible to stop at any time. The consent form used is anonymized and presented in Appendix A.6. This study was conducted for research purposes only.

## References


Hala Al Kuwatly, Maximilian Wich, and Georg Groh. 2020. Identifying and measuring annotator bias based on annotators' demographic characteristics. In *Proc. of the Fourth Workshop on Online Abuse and Harms*, pages 184–190. Association for Computational Linguistics.

Sajid Ali, Tamer Abuhmed, Shaker El-Sappagh, Khan Muhammad, Jose M. Alonso-Moral, Roberto Confalonieri, Riccardo Guidotti, Javier Del Ser, Natalia Díaz-Rodríguez, and Francisco Herrera. 2023. Explainable Artificial Intelligence (XAI): What we know and what is left to attain Trustworthy Artificial Intelligence. *Information Fusion*, page 101805.

Alejandro Barredo Arrieta, Natalia Díaz-Rodríguez, and Del Ser. 2020. Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI. *Information Fusion*, 58:82–115.

Milan Bhan, Nina Achache, Victor Legrand, Annabelle Blangero, and Nicolas Chesneau. 2023a. Evaluating self-attention interpretability through human-grounded experimental protocol. In *Proc. of the First World Conf. on Explainable Artificial Intelligence xAI*, pages 26–46.

Milan Bhan, Jean-noel Vittaut, Nicolas Chesneau, and Marie-jeanne Lesot. 2023b. Enhancing textual counterfactual explanation intelligibility through counterfactual feature importance. In *Proceedings of the 3rd Workshop on Trustworthy Natural Language Processing (TrustNLP 2023)*, pages 221–231, Toronto, Canada. Association for Computational Linguistics.

Milan Bhan, Jean-Noël Vittaut, Nicolas Chesneau, and Marie-Jeanne Lesot. 2023c. Tigtec: Token importance guided text counterfactuals. In *Proc. of the European Conf. on Machine Learning ECML-PKDD*, page 496–512. Springer.

Sergio Andrés Castaño-Pulgarín, Natalia Suárez-Betancur, Luz Magnolia Tilano Vega, and Harvey Mauricio Herrera López. 2021. Internet, social media and online hate speech. systematic review. *Aggression and violent behavior*, 58:101608.

Pierre Colombo, Marine Picot, Nathan Noiry, Guillaume Staerman, and Pablo Piantanida. 2023. Toward stronger textual attack detectors. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 484–505.

Darren Cook, Miri Zilka, Heidi DeSandre, Susan Giles, and Simon Maskell. 2023. Protecting children from online exploitation: Can a trained model detect harmful communication strategies? In *Proc. of the 2023 AAAI/ACM Conference on AI, Ethics, and Society*, AIES '23, page 5–14. Association for Computing Machinery.

David Dale, Anton Voronov, and Daryna Dementieva. 2021. Text Detoxification using Large Pre-trained Neural Models. In *Proc. of the 2021 Conf. on Empirical Methods in Natural Language Processing*, pages 7979–7996. Association for Computational Linguistics.

Xiaoli Fern and Quintin Pope. 2021. Text Counterfactuals via Latent Optimization and Shapley-Guided Search. In *Proc. of the 2021 Conf. on Empirical Methods in Natural Language Processing*, pages 5578–5593. Association for Computational Linguistics.

Paula Fortuna and Sérgio Nunes. 2018. A survey on automatic detection of hate speech in text. *ACM Comput. Surv.*, 51(4).

Gemma Gemma Team, Thomas Mesnard, Cassidy Hardin, Dadashi, et al. 2024. Gemma: Open models based on gemini research and technology. *arXiv preprint arXiv:2403.08295*.

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

Riccardo Guidotti. 2022. Counterfactual explanations and how to find them: literature review and benchmarking. *Data Mining and Knowledge Discovery*.

Skyler Hallinan, Alisa Liu, Yejin Choi, and Maarten Sap. 2023. Detoxifying Text with MaRCo: Controllable Revision with Experts and Anti-Experts. In *Proc. of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 228–242. Association for Computational Linguistics.

Zhiqiang Hu, Roy Ka-Wei Lee, Charu C. Aggarwal, and Aston Zhang. 2022. Text style transfer: A review and experimental evaluation. *SIGKDD Explor. Newsl.*, 24(1):14–45.

F. Jelinek, R. L. Mercer, L. R. Bahl, and J. K. Baker. 2005. Perplexity—a measure of the difficulty of speech recognition tasks. *The Journal of the Acoustical Society of America*, 62(S1):S63.

Shagun Jhaver, Alice Qian Zhang, Quan Ze Chen, Nikhila Natarajan, Ruotong Wang, and Amy X. Zhang. 2023. Personalizing content moderation on social media: User perspectives on moderation choices, interface design, and labor. *Proc. ACM Hum.-Comput. Interact.*, 7(CSCW2).

Jigsaw. 2018. Toxic comment classification challenge. `https://www.kaggle.com/c/jigsaw-toxic-comment-classification-challenge`. Accessed: 2010-09-30.

Thibault Laugel, Marie-Jeanne Lesot, Christophe Marsala, Xavier Renard, and Marcin Detyniecki. 2019. The Dangers of Post-hoc Interpretability:

Unjustified Counterfactual Explanations. In *Proc. of the Twenty-Eighth Int. Joint Conf. on Artificial Intelligence*, pages 2801–2807. Int. Joint Conf. on Artificial Intelligence Organization.

Léo Laugier, John Pavlopoulos, Jeffrey Sorensen, and Lucas Dixon. 2021. Civil rephrases of toxic texts with self-supervised transformers. In *Proc. of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1442–1461.

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 *Proc. of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

Hongshuo Liang, Erlu He, Yangyang Zhao, Zhe Jia, and Hao Li. 2022. Adversarial attack and defense: A survey. *Electronics*, 11(8):1283.

Scott M. Lundberg and Su-In Lee. 2017. A unified approach to interpreting model predictions. In *Proc. of the 31st Int. Conf. on Neural Information Processing Systems*, NIPS'17, pages 4768–4777.

Nishtha Madaan, Srikanta Bedathur, and Diptikalyan Saha. 2022. Plug and Play Counterfactual Text Generation for Model Robustness. ArXiv:2206.10429 [cs].

Vivek Miglani, Aobo Yang, Aram Markosyan, Diego Garcia-Olano, and Narine Kokhlikyan. 2023. Using Captum to Explain Generative Language Models. In *Proc. of the 3rd Workshop for Natural Language Processing Open Source Software (NLP-OSS 2023)*, pages 165–173. Association for Computational Linguistics.

Tim Miller. 2019. Explanation in artificial intelligence: Insights from the social sciences. *Artificial Intelligence*, 267:1–38.

Christoph Molnar. 2020. *Interpretable Machine Learning*. Lulu.com.

Cicero Nogueira dos Santos, Igor Melnyk, and Inkit Padhi. 2018. Fighting offensive language on social media with unsupervised text style transfer. In *Proc. of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 189–194. Association for Computational Linguistics.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proc. of the 40th Annual Meeting on Association for Computational Linguistics*, ACL '02, page 311–318.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. 2019. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. 2011. Scikit-learn: Machine learning in python. *Journal of Machine Learning Research*, 12:2825–2830.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global vectors for word representation. In *Proc. of the 2014 Conf. on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543. Association for Computational Linguistics.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2018. Language Models are Unsupervised Multitask Learners.

Kyle Rapp. 2021. Social media and genocide: The case for home state responsibility. *Journal of Human Rights*, 20(4):486–502.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2019. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. In *Proc. of the 2019 Conf. on Empirical Methods in Natural Language Processing and the 9th Int. Joint Conf. on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992. Association for Computational Linguistics.

Alexis Ross, Ana Marasović, and Matthew Peters. 2021. Explaining NLP Models via Minimal Contrastive Editing (MiCE). In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 3840–3852, Online. Association for Computational Linguistics.

Joni Salminen, Hind Almerekhi, Milica Milenković, Soon-gyo Jung, Jisun An, and Haewoon Kwak. 2018. Anatomy of online hate: developing a taxonomy and machine learning models for identifying and classifying hate in online news media. In *Proc. of the Int. AAAI Conf. on Web and Social Media*, volume 12.

Maarten Sap, Swabha Swayamdipta, Laura Vianna, Xuhui Zhou, Yejin Choi, and Noah A. Smith. 2022. Annotators with attitudes: How annotator beliefs and identities bias toxic language detection. In *Proc. of the 2022 Conf. of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5884–5906, Seattle, United States. Association for Computational Linguistics.

Ruth Spence, Antonia Bifulco, Paula Bradbury, Elena Martellozzo, and Jeffrey DeMarco. 2023. The psychological impacts of content moderation on content moderators: A qualitative study. *Cyberpsychology: Journal of Psychosocial Research on Cyberspace*, 17(4).

Mukund Sundararajan, Ankur Taly, and Qiqi Yan. 2017. Axiomatic attribution for deep networks. In *Proc. of the 34th Int. Conf. on Machine Learning, ICML*, volume 70 of *ICML'17*, pages 3319–3328. JMLR.org.

Kurt Thomas, Devdatta Akhawe, Michael Bailey, Dan Boneh, Elie Bursztein, and Sunny Consolvo. 2021. Sok: Hate, harassment, and the changing landscape of online abuse. In *2021 IEEE Symposium on Security and Privacy (SP)*, pages 247–267.

Minh Tran, Yipeng Zhang, and Mohammad Soleymani. 2020. Towards a friendly online community: An unsupervised style transfer framework for profanity redaction. In *Proc. of the 28th Int. Conf. on Computational Linguistics*, pages 2107–2114. International Committee on Computational Linguistics.

Marcos Treviso, Alexis Ross, Nuno M. Guerreiro, and André Martins. 2023. CREST: A joint framework for rationalization and counterfactual text generation. In *Proc. of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15109–15126. Association for Computational Linguistics.

Joseph Walther. 2022. Social media and online hate. *Current Opinion in Psychology*, 45.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. 2020. Transformers: State-of-the-art natural language processing. In *Proc. of the Conf. on Empirical Methods in Natural language Processing: system demonstrations, EMNLP*, pages 38–45.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. 2019. Bertscore: Evaluating text generation with bert. In *Int. Conf. on Learning Representations*.

# A Appendix

## A.1 Scientific libraries

We used several open-source libraries in this work: pytorch (Paszke et al., 2019), HuggingFace transformers (Wolf et al., 2020) sklearn (Pedregosa et al., 2011) and Captum (Miglani et al., 2023). We will make our code available upon acceptance to facilitate reproduction and further research.

## A.2 Implementation Details

The code used to run the experiments is not provided for anonymity reasons and will be available upon acceptance.

**Pretrained language models** The library used to import pretrained language model to be finetuned to distinguish toxic and non-toxic content is `transformer`. In particular, the backbone version of BERT is `bert-base-uncased` the one of DistilBERT is `distilbert-base-uncased`. The library used to import the Sentence Transformer is `sentence_transformers` and the model backbone is `paraphrase-MiniLM-L6-v2`. The backbone of the pre-trained toxicity classifier is `toxic-bert`.

**Classifier performance** The classifier performance after training is respectively 94.1% and 90.2% for BERT and DistilBERT. This way, CF text detoxification is performed with classifiers that have been trained on a different dataset from the ones used for the evaluation.

**TIGTEC cost function** The cost function driving CF search is defined such as: $c(x_{\text{cf}}, x) = -(p(y_{\text{non-toxic}}|x_{\text{cf}}) - \alpha d_s(x_{\text{cf}}, x))$ with $d_s(x, x') = \frac{1}{2}(1 - s(x, x'))$ where $p(y_{\text{non-toxic}})$ is the probability score of being non-toxic with respect to the trained classifier, $s$ is a cosine similarity measure between text embeddings obtained from Sentence Transformers and $\alpha$ is a hyperparameter set to 0.3.

**Datasets** MAgr contains 425 toxic texts, SBF 557 and DynaHate 662. The three datasets are available in the Github project in the initial paper[2].

**Quality metrics** The perplexity score is computed based on the exponential average cross-entropy loss of Gemma-2B (Gemma Team et al., 2024), a recently developed small generative

[2] https://github.com/shallinan1/MarcoDetoxification/tree/main/datasets

language model outperforming GPT-2. The library to import the pre-trained model is `transformers` and the backbone is `gemma-2b`. Due to the presence of outliers when calculating the entropy used to calculate perplexity, aggregation is performed with the median rather than the mean operator.

### A.3 Ablation study

Table 3 shows the experimental results obtained by running three different versions of $\texttt{CF-Detox}_{\text{tigtec}}$ on the three datasets of interest. Each $\texttt{CF-Detox}_{\text{tigtec}}$ instance is defined by the LFI method used to target toxic POS. Self-attention is aggregated as in the original `TIGTEC` paper by averaging all the attention coefficients related to the CLS token over the attention heads in the last layer of the $f$ classifier. Table 3 shows that Self-attention, Integrated gradients and KernelSHAP lead to similar results in terms of toxicity mitigation, content preservation and text plausibility. These results highlight that LFI methods of a different nature (perturbation, attention or gradient-based) can all yield good results, which underpins $\texttt{CF-Detox}_{\text{tigtec}}$ robustness.

Toxicity mitigation through CF generation methods like `TIGTEC` has to be performed by choosing the appropriate LFI method to target toxicity based on the available model information. For example, KernelSHAP is appropriate if no information (internal parameters, gradients) is available about the classifier $f$ used to counterfactually mitigate toxicity, due to its *model-agnostic* nature. On the contrary, if $f$ gradients are accessible, the use of Integrated gradients is recommended since it is less computationally costly than KernelSHAP. Finally, if all $f$ parameters are accessible, using Self-attention is appropriate because it is available at no cost.

Since `TIGTEC` gradually masks and replaces tokens in the original toxic text based on LFI, we postulate that the sparser detoxified texts a LFI method induces, the better its performance, as it targets the most discriminating tokens of the initial text with respect to $f$. This way, Integrated gradients give the most *faithful* explanations (i.e. target the most accurately toxicity).

### A.4 Human-grounded experiment

Before running the experiment, annotators are given the same instructions (see the project presentation in Figure 6). To make sure that they annotate based on the same common knowledge, we define the textual toxicity as "*violent, aggressive or offensive language that may focus on a specific person or group of people sharing a common property. This common property can be gender, sexual orientation, ethnicity, age, religion or political affiliation.*". Annotators all have a MSc degree in data analytics or machine learning and have a good knowledge of English.

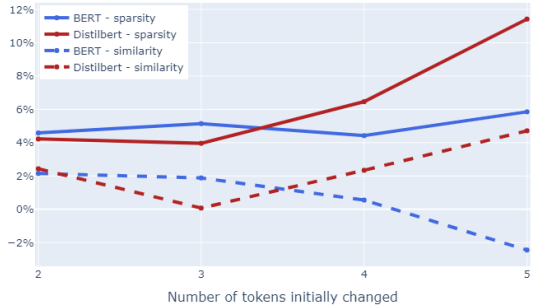


Figure 5: Evolution of the gain in sparsity and similarity induced by CFI on Dynahate, according to the number of tokens initially modified, per model.

### A.5 CFI enhancement

Figure 5 displays the relation between the gain induced by CFI and the number of tokens initially modified in the first version of the CF by focusing on the DynaHate dataset. The number of considered token modifications stops at 5, representing 95% of the CFs initially generated. It turns out that for DistilBERT, the higher the number of initially modified tokens, the higher the sparsity and proximity gain (up to +11% for sparsity). Results are overall more stable for BERT across the number of tokens initially changed.

### A.6 Human-grounded protocol participant consent form

Each participant signed an informed consent form containing the project purpose and details and the intended use of the data they would generate. Figure 6 shows an anonymized version of the consent form.

| Dataset | Metric | CF-Detox$_{tigtec}$ | | | | | |
|---|---|---|---|---|---|---|---|
| | | **Bert** | | | **Distilbert** | | |
| | | attention | IG | kshap | attention | IG | kshap |
| Dynahate | ΔPPL ↓ | 1.34 | 1.26 | **1.25** | 1.43 | **1.24** | 1.25 |
| | %CP ↑ | **87.4** | 86.3 | 86.4 | **89.2** | 87.0 | 86.5 |
| | %S ↑ | 86.8 | **87.4** | 86.1 | **87.4** | **87.4** | 86.3 |
| | %ACC ↑ | 81.6 | **84.0** | 82.9 | 79.0 | 81.4 | **81.9** |
| | SCORE ↓ | 0.23 | **0.20** | 0.21 | 0.27 | 0.23 | **0.23** |
| MAgr | ΔPPL ↓ | 2.38 | 2.14 | **2.04** | 2.76 | 2.26 | **2.16** |
| | %CP ↑ | **88.0** | 85.8 | 84.6 | **90.5** | 87.1 | 86.1 |
| | %S ↑ | **70.7** | **70.7** | 70.1 | **71.4** | 70.8 | 70.1 |
| | %ACC ↑ | 94.4 | **95.5** | 95.1 | 89.0 | **92.0** | 91.8 |
| | SCORE ↓ | **0.13** | 0.10 | 0.10 | **0.19** | 0.14 | 0.14 |
| SBF | ΔPPL ↓ | 1.35 | **1.26** | 1.27 | 1.32 | 1.27 | **1.26** |
| | %CP ↑ | **91.5** | 90.4 | 90.2 | **92.3** | 90.9 | 90.4 |
| | %S ↑ | 88.7 | **89.2** | 87.6 | 89.0 | **89.3** | 88.4 |
| | %ACC ↑ | 86.4 | 86.5 | **87.6** | 82.4 | 84.0 | **84.2** |
| | SCORE ↓ | 0.19 | 0.17 | **0.16** | 0.24 | **0.21** | **0.21** |

Table 3: CF-Detox$_{tigtec}$ counterfactual toxicity mitigation by LFI POS toxicity targeting method.

**censored- INFORMED CONSENT FORM**

**Project Title** : Evaluation of different text detoxification methods

**Project Leaders** : **censored**

**Research location** : **censored**

**Project Presentation**

The detection and treatment of toxicity is an important topic in the field of natural language processing. There are different methods for identifying toxic terms in a text and modifying them. We have developed a new method based on the counterfactual technique which we wish to compare to existing methods. However, the performance of these different methods is difficult to evaluate. A human evaluation seems the most relevant approach at our disposal.

If you agree to participate in this study, we will ask you to evaluate whether short texts are toxic or not or to rank them from most toxic to least toxic. The approximate duration of the assessment is around thirty minutes.

**Your privacy rights**

All the information collected during this experiment for the pursuit of the purposes set out in the previous paragraph will be processed by **censored**, anonymously and will remain confidential. The legal basis for processing is your consent.

These will be kept in a computer file that complies with the applicable regulations in force (General Data Protection Regulations and Data Protection Act).

The data collected will be communicated only to the following recipients:

- **censored**,
- **censored**,
- **censored**,
- **censored**

The results obtained from the processing of this questionnaire may be the subject of scientific publications, but the identity of the participants will not be revealed, and no information that could reveal your identity will be disclosed.

The data is kept until the publication of an article or a maximum of 3 years.

**Your rights to withdraw from this research at any time**

Participation in this study is completely voluntary. Please note that even if you decide to complete this questionnaire, it is possible to stop completing it at any time, and as long as the final registration has not been made, none of your data will be processed.

You can access the data concerning you, rectify it, request its deletion or exercise your right to limit the processing of your data. You can withdraw your consent to the processing of your data at any time; you can also object to the processing of your data. Visit the cnil.fr website for more information on your rights.

To exercise these rights, you can contact **censored**

If you believe, after contacting us, that your "Data Protection" rights are not respected, you can file a complaint with the CNIL.

**Diffusion**

The results of this research may be published in scientific journals or be the subject of communications at scientific conferences.

You can ask questions about the research at any time by contacting the project managers by email **censored**

**Consent to participate**

**By checking the box** below and **signing this consent form**, you certify that you have read and understood the above information and that you have been informed of your right to withdraw your consent or withdraw from this research at any time, without prejudice.

☐ ***I have read and understood the above information and I voluntarily agree to participate in this research.***

**Done at** : _____

**On the** : _____

**Name, First Name** : _____

**Signature** :

Figure 6: Human-grounded protocol participant consent form