计算机系统应用 ISSN 1003-3254, CODEN CSAOBN
[doi: 10.15888/j.cnki.csa.010111] [CSTR: 32024.14.csa.010111]

E-mail: csa@iscas.ac.cn
http://www.c-s-a.org.cn
Tel: +86-10-62661041

面向隐式毒性的多阶段多目标文本去毒①

毛　柯[1,2], 安俊秀[1,2], 王　鑫[1,2], 袁明坤[1,2]

[1](成都信息工程大学 软件工程学院, 成都 610225)
[2](成都信息工程大学 并行计算与大数据研究所, 成都 610225)
通信作者: 安俊秀, E-mail: anjunxiu@cuit.edu.cn

摘　要: 针对现有文本去毒方法未充分考虑隐式毒性以及去毒后文本质量较差的问题, 提出一种多阶段多目标优化的文本去毒框架 MSMO-Detox (multi-stage multi-objective detoxification). 本方法采用 3 阶段级联处理实现精准去毒: 首先运用基于标记的毒性解释技术, 通过传播分解向量以精确识别毒性贡献度超阈值的词元并进行掩码处理; 随后采用专家乘积 (product of experts, PoE) 框架进行词元生成, 替换掉被掩码词元; 最后实施多目标重排序策略, 从隐式毒性、文本流畅度、语义保留这 3 个维度综合评估候选句子, 选取评分最优的候选句作为输出. 实验结果表明, 在 MAgr、SBF、DynaHate、Jigsaw 数据集上, MSMO-Detox 相较于不同数据集上的最优基线方法, 毒性指标分别平均下降 23.1%、23.9%、17.6%、5.6%, 此外, 文本流畅度与语义保留能力也得到改善. 可见, MSMO-Detox 在文本去毒任务中具有显著优势, 特别是在网络生态优化中, 该方法可以作为网络生态优化中去除网络暴力的重要工具, 用于有毒文本的风格迁移.

关键词: 文本去毒; 隐式毒性; 文本风格迁移; 多目标优化; 网络生态治理

引用格式: 毛柯,安俊秀,王鑫,袁明坤.面向隐式毒性的多阶段多目标文本去毒.计算机系统应用. http://www.c-s-a.org.cn/1003-3254/10111.html

Multi-stage Multi-objective Text Detoxification for Implicit Toxicity

MAO Ke[1,2], AN Jun-Xiu[1,2], WANG Xin[1,2], YUAN Ming-Kun[1,2]

[1](School of Software Engineering, Chengdu University of Information Technology, Chengdu 610225, China)
[2](Institute of Parallel Computing and Big Data, Chengdu University of Information Technology, Chengdu 610225, China)

Abstract: Existing textual detoxification methods do not fully consider implicit toxicity, and detoxified text often has low quality. To address these problems, a multi-stage multi-objective detoxification framework, termed as MSMO-Detox, is proposed. MSMO-Detox uses a three-stage cascade for precise detoxification. First, a marker-based toxicity attribution technique propagates decomposition vectors to identify tokens whose toxicity contribution exceeds a threshold and performs masking on these tokens. Second, a product of experts (PoE) framework generates replacement tokens for masked positions. Third, a multi-objective reranking strategy conducts a comprehensive evaluation of candidate sentences across implicit toxicity, fluency, and semantic preservation, and selects the highest-scoring candidate as the output. Experimental results show that on MAgr, SBF, DynaHate, and Jigsaw datasets, MSMO-Detox reduces toxicity metrics by an average of 23.1%, 23.9%, 17.6%, and 5.6%, compared with the best baseline on each dataset. Fluency and semantic preservation also improve. MSMO-Detox demonstrates clear advantages in textual detoxification and can be applied to the task of toxic-text style transfer as an important tool for the elimination of cyber violence and optimizing online ecosystems.

Key words: text detoxification; implicit toxicity; text style transfer; multi-objective optimization; network ecological governance

① 基金项目: 国家社会科学基金 (22BXW048)
收稿时间: 2025-09-11; 修改时间: 2025-10-10; 采用时间: 2025-10-29; csa 在线出版时间: 2026-03-02

网络生态快速演进带来高度的互动性与参与性, 信息过载、内容低质等挑战随之涌现. 低质内容与暴力语言蔓延尤为突出, 对网络空间健康发展构成严重威胁[1–3]. 文本去毒 (text detoxification) 作为自然语言处理领域中的一项重要任务, 致力于识别并修正含有显式或隐式有害内容的自然语言表达, 以强化网络空间生态治理[4,5]. 然而现有去毒方法主要聚焦显式毒性检测与去除, 对文本质量关注不足, 从而导致在修改有害内容方面存在明显局限性.

现有文本去毒策略涵盖基于规则、检索式、基于生成、对抗生成等多种方法. Bhan 等[6]依赖预定义毒性词典以及正则表达式, 对输入文本中的有毒词语实施屏蔽、替换或删除, 但是该方法难以识别隐晦和存在上下文依赖的毒性, 仅能处理显式脏话, 面对谐音以及语境表达的毒性则束手无策. Dementieva 等[7]采用“删除-检索-生成”框架, 先删除文本有毒标志词, 再从无毒语料或替换词库检索相似关键词, 最终生成去毒版本. 然而该方法依赖高质量的检索语料库, 在没有近似无毒关键词时效果较差, 并且对上下文理解与多义词辨析能力有限. 另外, 一些著名的架构使用神经网络生成方法, 通过训练生成模型直接将有毒文本重写为无毒文本, Pour 等[8]通过对比学习和非似然损失显著提升了效果, 但是该类风格迁移方法往往损失部分原句细节, 导致语义保留性较差.

针对上述局限, 本文设计出一种多阶段处理与多目标优化的文本去毒方法. 该方法融合隐式毒性去毒与文本质量处理, 集成能有效处理细微毒性的专家乘积模型和 ToxiGen-HateBERT[9]毒性检测模块. 将多目标重排序策略引入其中, 运用多项评价指标对生成文本进行综合考量, 有效规避语义漂移与非自然表达.

本文核心贡献如下.

(1) 设计基于 3 段流水线的隐式毒性去毒框架: 采用“识毒掩码-填充掩码-重排选优”的级联架构, 最终实现将潜在有毒令牌精确掩码、填充掩码并使用多指标筛选最优去毒文本的端到端流程.

(2) 集成隐式毒性处理机制: 运用能处理细微毒性的专家乘积模型, 将 ToxiGen-HateBERT 毒性检测模块整合至重排序阶段. 使得对微妙偏见、隐含攻击性内容的识别与处理能力得到显著提升.

(3) 建立多目标优化体系: 兼顾毒性抑制、文本流畅度与语义保留, 在降低毒性风险同时有效规避传统方法普遍存在的过度修改问题.

1 相关工作

1.1 文本去毒

文本去毒致力于将含有有害或攻击性内容的文本转化为去除毒性的文本, 并且尽可能保留原始含义, 通常被视作风格迁移问题[10]. 早期文本去毒方法主要依赖简单词汇替换和规则匹配, 但这些方法常导致语义和流畅度下降. 随着深度学习技术发展, 基于神经网络的文本去毒方法逐渐成为主流. Dale 等[11]采用预训练语言模型进行文本去毒, 训练分类器识别毒性词汇并实施掩码处理, 但在生成多样性方面受限. Feng 等[12]设计 DuNST 方法为每个样本生成毒/非毒伪标注, 训练中加入随机噪声干扰, 引导模型更好地区分并避免生成有毒文本, 能显著降低毒性, 对流畅性和创新性影响较小, 但通常依赖大规模训练模型和精心设计控制信号, 对训练数据和架构有一定要求. 同时, 扩散模型因其生成多样样本的能力, 近期也被引入文本去毒领域. Lee 等[13]提出的 XDetox 方法为有毒文本产生多个去毒版本, 增加选择空间, 但是在重排序阶段使用指标过于单一, 容易产生不连贯文本. Floto等[14]提出了一种混合条件与非条件的文本扩散模型, 能够产生多种去毒版本, 增加选择空间, 但是训练数据需求大、训练成本高、在数据不足时容易产生不连贯文本.

1.2 多目标重排序策略

多目标优化与候选重排序已经成为文本生成领域的关键范式. 多目标方法超越了单一指标的优化, 发展出能平衡流畅性、语义保真、属性控制等相互冲突目标的复杂框架, 与重排序相互结合生成多候选并以多评估器综合打分实现工程化权衡. 端到端的多目标方法, 比如可控偏好优化、Pareto 探索等计算代价高、超参调优困难[15–17]. 近来, 在 PAN/CLEF 的多语言去毒任务中, 参赛系统普遍采用“候选生成+多指标打分”的通道, 以确保跨语言场景下的语义保真与流畅度[18]. Řehulka 等[19]提出的 RAG meets detox 通过检索相关上下文生成候选, 再结合重排序器实现安全与保真的平衡. 因此, 将这两类方法融合成为近期研究倾向.

2 本文方法

本文模型总体框架如图 1 所示, 主要涉及 3 个部分: 毒性词元识别与掩码、基于 MaRCo[20]的掩码填充

与多目标重排序.

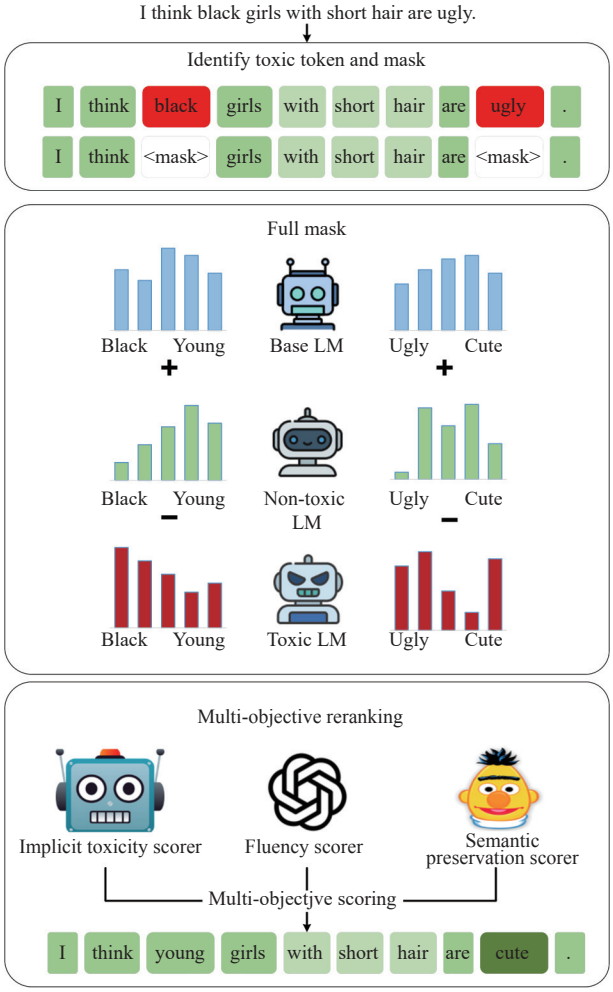

图 1 MSMO-Detox 总体框架

2.1 毒性词元识别与掩码

DecompX[21]是一种基于分解传播的 Transformer 决策解释方法, 核心思想是构建分解的令牌表示并在模型中逐层传播而不在层间混合. 对于输入文本的每个令牌, 该方法维护独立的向量表示, 并通过整个网络跟踪其对最终预测的贡献. 在此任务中, 利用 DecompX 识别输入文本中对毒性预测贡献最大的词元. 获得毒性词元的公式如下:

$$Importance(x_j)=\sum_{c=1}^{C} y_{c\leftarrow x_j} \tag{1}$$

其中, 计算累积重要度分数$Importance(x_j)$量化每个词元x_j在所有类别 C 中对特定类别 c (例如毒性、情感) 的重要性, 并用$y_{c\leftarrow x_j}$表示词元x_j对类别 c 预测分数的贡献度.

DecompX 本身只是一种可解释性技术, 并不直接参与文本去毒或生成, MSMO-Detox 将 DecompX 的解释结果主动引入去毒流程, 实现对去毒任务中隐式毒性的主动定位, 是赋予 DecompX 的应用层面创新.

2.2 基于 MaRCo 的掩码填充

识别并掩码毒性词元后, 需要使用合适的非毒性词元进行替换, 本文采用 MaRCo 方法进行高质量的非毒性替换, 该方法利用专家模型和反专家模型的对比来生成高质量的替换词.

MaRCo 采用专家乘积 (product of experts, PoE) 框架进行词元生成, 给定掩码后的序列w^m, 掩码标记填充的公式为:

$$P(X_i \mid g_{<i}, w, w^m)=Softmax\left(z_i+\alpha_1 z_i^{+}-\alpha_2 z_i^{-}\right) \tag{2}$$

其中, X_i是一个在词汇表 V 上的随机变量, 表示在给定先前生成的上下文$g_{<i}$情况下, 文本序列中第 i 个掩码位置预测的非毒性替换标记, w和w^m分别是原始句子和被掩码的句子, 分别用于指导基础模型以及专家和反专家模型进行替换选择. z_i、z_i^{+}、z_i^{-}为分别来自基础模型、非毒性模型和毒性模型的 logits (未归一化分数). 超参数α_1与α_2分别独立控制专家和反专家的影响, 以实现最佳替换效果.

MaRCo 本身也能对有害词元进行掩码处理, 但是其掩码能力远不及填充生成的能力, 因为在掩码过程中 MaRCo 方法并未考虑决策过程的问题. 为此 MSMO-Detox 将毒性词元的识别与填充生成进行明确的分离, 使得掩码与填充过程既准确又可靠.

2.3 多目标重排序机制

经过掩码填充后, 将生成的多个候选文本进行综合评分, 评分过程中综合考虑隐式毒性、文本流畅度、语义保留这 3 个维度. 最后按照评分高低进行重排序以选择最优输出.

对于候选文本$X=\{x_1, x_2, \cdots, x_k\}$, 采用多目标优化框架对每个候选文本$x_i$计算综合评分:

$$Score(x_i)=\lambda_1(1-T(x_i))+\lambda_2 F(x_i)+\lambda_3 S(x_i) \tag{3}$$

其中, λ_1、λ_2、λ_3为各目标的权重系数, 满足$\sum_{j=1}^{3}\lambda_j=1$. $T(x_i)$为 ToxiGen-HateBERT 模型预测候选句子的毒性概率, 数值越大表明毒性越强, 为了确保毒性越小的候选语句得分越高, 采用取反操作; $F(x_i)$为流畅度评分;

$S(x_i)$为语义保留评分.

2.3.1 隐式毒性 (Toxicity) 评分

传统的毒性检测器可能忽略隐含的偏见和刻板印象. 本文引入专门针对隐含偏见和微妙有害内容的 ToxiGen-HateBERT 模型, 其专门针对隐式毒性检测进行了微调, 因此对细微的毒性表达具有更强的识别能力.

对于候选句子x_i, ToxiGen-HateBERT 输出两个分类器的 logits (分别对应着“非毒性”与“毒性”), 记为z_{nt}, z_t. 通过 *Softmax* 将 logits 转换为概率:

$$T(x_i)=\frac{\exp(z_t)}{\exp(z_t)+\exp(z_{nt})} \tag{4}$$

其中, $T(x_i)$即为模型预测候选句子为“毒性”类别的概率. 采用 *Softmax* 输出概率可以保证$T(x_i)\in(0,1)$.

2.3.2 流畅度 (Perplexity) 评分

利用 GPT-2 XL 语言模型计算文本困惑度, 并归一化为流畅度分数. 对于给定句子$x_i=\{x_{i1},x_{i2},\cdots,x_{iT}\}$, 困惑度定义为预测下一个词的平均概率的倒数指数:

$$PPL(x_i)=\exp\left(-\frac{1}{T}\sum_{j=1}^{T}\log p_\theta(x_{ij}|x_{i<j})\right) \tag{5}$$

其中, $p_\theta(x_{ij}|x_{i<j})$为 GPT-2 XL 对候选句x_i的第 j 词的预测概率, *PPL* 值越低表示该候选句越流畅.

由于原始困惑度可能存在数值范围较广, 并且随着长度不同而变化较大, 为此将数值进行最大最小归一化处理. 直接做归一化操作时, 极端大的值会让候选句子压缩到接近于 0, 导致分布极其不均匀, 为此先将困惑度取对数:

$$L(x_i)=\log PPL(x_i) \tag{6}$$

随后得到同一输入的候选集上的最小值$P_{\min}$和最大值$P_{\max}$, 并做最大最小归一化:

$$L_{\text{norm}}(x_i)=\frac{L_{\text{norm}}(x_i)-P_{\min}}{P_{\max}-P_{\min}} \tag{7}$$

最后取反得到流畅度评分$F(x_i)$:

$$F(x_i)=1-L_{\text{norm}}(x_i) \tag{8}$$

2.3.3 语义保留评分 (BERTScore)

语义保留评分旨在确保去毒后的文本保持原始语义的完整性. 该指标通过提取候选句子x_i与参考句子 R 中每个词 (或者词片段) 对应的上下文嵌入$\{c_i\}$、$\{r_j\}$, 然后计算两者之间的相似度并进行最大匹配. 将候选词匹配到参考中相似度最高的词来计算 *Precision*, 将参考词匹配到候选中相似度最高的词来计算 *Recall*, 语义覆盖度与原始词的语义保留度的计算公式如下:

$$Precision=\frac{1}{|C|}\sum_{i=1}^{|C|}\max_j\cos(c_i,r_j) \tag{9}$$

$$Recall=\frac{1}{|R|}\sum_{j=1}^{|R|}\max_i\cos(c_i,r_j) \tag{10}$$

其中, $|C|$与$|R|$表示候选句子x_i和参考句子中的 token 总数. 最后根据 *Precision* 与 *Recall* 的值来计算两者的调和平均数得到x_i的语义保留评分$S(x_i)$, 其取值范围为[0, 1], 值越大表明语义保留越好. 计算公式如下所示:

$$S(x_i)=\frac{2Precision\cdot Recall}{Precision+Recall} \tag{11}$$

3 实验

3.1 数据集

为验证本文模型的性能, 采用了 4 个此前在去毒任务中使用过的不同数据集, 如表 1 所示, 数据集介绍如下.

表 1 数据集详细信息

数据集	Validation	Test
SBF	92	114
MAgr	238	298
DynaHate	1858	2011
Jigsaw	—	10000

SBF (social bias frames) 数据集[22]在构建的过程中选取了社交媒体上包含隐式偏见内容的帖子子集, 尤其是来自“微歧视”子版块的样本. 这些语料往往是不显眼的种族、性别和其他社会偏见隐喻, 需要综合语义才能判断是否含有毒性.

MAgr 数据集来自一个名为 Microaggressions.com 的 Tumblr 博客. 该博客允许用户匿名发布带有社会偏见的言论或事件描述, 每条帖子通常包含一个冒犯性的引用语句或事件描述, 是一个真实场景下的微歧视语言语料.

DynaHate 数据集[23]是一个动态生成的仇恨言论数据集, 采用人机迭代式的方法进行构建, 其中包含复杂隐晦的仇恨表达.

Jigsaw 数据集是 Google Jigsaw 团队发布的“有毒评论分类”数据集, 来自 Kaggle 的一项公开挑战“Unintended Bias in Toxicity Classification”, 原始语料来自

CivilComments 平台的用户评论 (如新闻评论区). 旨在尽量减少与身份相关的意外模型偏差.

3.2 评价指标

在文本去毒生成任务中, 评价生成结果的质量需同时兼顾多个维度, 包括毒性、语义保留、上下文流畅度和语义相似度等, 本文选取在先前研究中均有运用的 4 种常用评估指标来衡量模型的性能.

使用 Perspective API 来衡量毒性, Google Perspective API 对生成句子返回取值范围在 (0, 1) 之间的安全性评价指标, 分数越低表示文本中“有毒”内容越少, 生成结果越安全可靠. 随后使用 BLEU (bilingual evaluation understudy) 和 BERTScore 指标来衡量生成文本相对于参考文本的保留能力, BLEU 关注的是表面形式的保留程度、BERTScore 则关注深处语义的保留程度, 两者的结合可以更全面地评估文本重写任务中的内容保留能力. 最后引入 Perplexity 指标, 利用 GPT-2 XL 评价生成文本的流畅度, Perplexity 值越小表明越符合语言模型的统计分布, 流畅度和可读性越高.

3.3 基线模型

本文工作主要侧重于文本去毒中的隐式毒性去毒, 为此, 选择相关基线模型作为比较标准. 使用 3 个该领域较为先进的模型, 包括 ParaGeDi[11]、CondBERT[11] 以及 MaRCo[20]. 通过与这些基线模型的比较, 能够有力地证实 MSMO-Detox 在去毒和提高文本质量上的优势.

ParaGeDi: 使用释义模型生成同义句, 但是通过风格条件语言模型控制输出, 来达到避免生成毒性词汇的目的.

CondBERT: 通过逻辑回归分类器计算词语的“毒性分数”, 随后将分数高于阈值的词汇标记为“毒性词汇”, 最后再用 BERT 生成非毒性同义词替换.

MaRCo: 使用 Product of Experts 框架与自编码器语言模型来达到去毒效果, 该模型专门用于识别并修正文本中的毒性内容.

3.4 实验环境与参数设置

3.4.1 实验环境

实验运行在 Ubuntu 20.04 操作系统上, 中央处理器为 10 vCPU Intel Xeon Processor (Skylake, IBRS), GPU 为 NVIDIA A100-PCIE-40GB, 模型代码在 Python 3.8 中使用 PyTorch 2.0.0 开发实现.

3.4.2 掩码超参数

确定掩码超参数时需要实现文本在毒性、流畅性、内容保留方面的最佳平衡, 为此将所有数据集在{0.1, 0.15,0.20,⋯,0.5}范围内进行联合搜索, 最终确定使用如表 2 所示的参数. Jigsaw 被定为高毒性数据集, SBF 与 MAgr 数据集毒性则较为微妙, 从联合搜索得到的阈值也证明了此点. 另外考虑到 Jigsaw 数据集中包含着大量较长句子, 需要更多的 token 参与解释并进行掩码处理, 便将 Batch size 设置得较小以避免显存超载.

表 2 掩码超参数信息

数据集	掩码阈值	Batch size
SBF	0.25	25
MAgr	0.25	25
DynaHate	0.2	25
Jigsaw	0.15	10

3.4.3 填充过程超参数

在填充过程中, 借鉴了 MaRCo 发布的微调模型, 具体参数设置如表 3 所示.

表 3 填充过程超参数信息

参数	SBF	MAgr	DynaHate	Jigsaw
反专家模型权重系数	1.5	1.5	1.5	1.5
专家模型权重系数	5.0	4.25	4.75	4.75
Repetition penalty	1.5	1.0	1.0	1.0
Temperature (base model)	2.9	2.5	2.5	2.5
Batch size	25	25	25	10

3.4.4 多目标重排序过程超参数

在多目标重排序的过程中, 将毒性抑制程度放在首位, 随后考虑生成文本的流畅性, 最终采用网格搜索的方式确定能带来最佳评分的各个权重值. 对于不同的数据集, 令 $\lambda_1 \in \{0.4,0.45,\cdots,0.6\}$、$\lambda_2 \in \{0.25,0.3,0.35,0.4\}$, 并以 $\lambda_3=1-\lambda_1-\lambda_2$ (同时满足 λ_3 不为 0 的组合) 构成候选权重集合进行搜索, 最终得到表 4 所示超参数.

表 4 多目标重排序过程超参数信息

数据集	λ_1	λ_2	λ_3
SBF	0.50	0.35	0.15
MAgr	0.50	0.35	0.15
DynaHate	0.50	0.35	0.15
Jigsaw	0.45	0.3	0.25

同时对于不同的数据集, 统一将候选句子的数量设置为 5, 从 5 个候选句中选出综合评分最高的句子作为输出.

3.5 实验结果

实验结果如表 5 所示, 展示了本文提出的 MSMO-Detox 方法与其他基线方法在 MAgr、SBF、DynaHate、Jigsaw 数据集上的实验结果, 其中粗体表示最佳结果,

下划线表示次优结果.

表 5 不同模型对比实验

数据集	Method	Validation				Test			
		Toxicity	Perplexity	BERTScore	BLEU	Toxicity	Perplexity	BERTScore	BLEU
MAgr	Original	0.280	52.13	—	—	0.258	70.19	—	—
	CondBERT	0.170	179.65	0.937	0.689	0.152	160.20	0.939	0.687
	ParaGeDi	0.148	124.11	0.922	0.461	0.151	113.56	0.925	0.450
	MaRCo	0.143	43.55	0.957	0.768	0.139	**39.33**	**0.954**	0.751
	MSMO-Detox	**0.115**	**37.42**	**0.960**	**0.785**	**0.102**	40.23	0.953	**0.769**
SBF	Original	0.349	58.46	—	—	0.342	88.79	—	—
	CondBERT	0.219	137.29	0.931	0.661	0.205	115.71	0.938	0.693
	ParaGeDi	0.171	188.67	0.913	0.392	0.176	103.81	0.924	0.467
	MaRCo	0.177	54.98	0.947	0.731	0.178	48.56	0.948	0.709
	MSMO-Detox	**0.131**	**47.18**	**0.954**	**0.745**	**0.133**	**42.85**	**0.956**	**0.750**
DynaHate	Original	0.536	205.76	—	—	0.555	222.55	—	—
	CondBERT	0.291	254.05	0.941	**0.725**	0.293	271.21	0.940	**0.735**
	ParaGeDi	0.289	221.46	0.915	0.468	0.211	341.28	0.892	0.282
	MaRCo	0.259	**110.36**	0.939	0.707	0.261	127.92	0.936	0.686
	MSMO-Detox	**0.191**	141.01	**0.947**	0.721	**0.192**	**114.35**	**0.945**	0.716
Jigsaw	Original	—	—	—	—	0.738	364.71	—	—
	CondBERT	—	—	—	—	0.198	288.72	**0.939**	0.691
	ParaGeDi	—	—	—	—	0.229	309.52	0.894	0.392
	MaRCo	—	—	—	—	0.296	**166.41**	0.926	0.652
	MSMO-Detox	—	—	—	—	**0.187**	188.60	0.936	**0.692**

由表 5 可以得出, MSMO-Detox 在 4 个数据集上均取得了最优的去毒效果. 具体而言, MSMO-Detox 与最优基线相比, 在数据集 MAgr、SBF、DynaHate、Jigsaw 上, 毒性值分别平均降低了 23.1%、23.9%、17.6%、5.6%. 其中 Jigsaw 是显式毒性数据集, 从侧面印证出 MSMO-Detox 对于显式毒性也能起到较好的毒性抑制效果. 值得注意的是, 在取得良好去毒效果的同时, MSMO-Detox 保持了良好的文本质量. 与各个基线模型相比, Perplexity、BERTScore、BLEU 评分均能达到最优或次优的水平. 综合各项指标, MSMO-Detox 成功实现了毒性抑制与文本质量的优化平衡, 验证了多阶段多目标优化框架的有效性.

为更加直观地呈现 MSMO-Detox 方法的实际去毒效果, 通过表 6 展示了该方法在 4 个不同数据集上的去毒后的文本示例.

表 6 去毒示例

数据集	操作	内容
MAgr	Original	You have so many black friends.
	Masking	You have so many <mask> friends.
	Results	You have so many good friends.
SBF	Original	Wow. She is actually quite beautiful for a Samoan.
	Masking	Wow. She is actually quite beautiful for a <mask>.
	Results	Wow. She is actually quite beautiful for a woman!
Dyna-Hate	Original	My white laptop is better than all white women.
	Masking	My <mask> laptop is better than all <mask> <mask>.
	Results	My stylish laptop is better than all the others.
Jigsaw	Original	Is it possible that you are as stupid as you sound?
	Masking	Is it possible that you are as <mask> as you sound?
	Results	Is it possible that you are as confused as you sound?

3.6 消融实验

为了评估多目标重排序机制对该文本去毒方法的性能影响, 将模型在有无多目标重排序组件的情况下分别进行实验, 表 7 记录了最终的实验结果.

表 7 消融实验

数据集	Method	Validation				Test			
		Toxicity	Perplexity	BERTScore	BLEU	Toxicity	Perplexity	BERTScore	BLEU
MAgr	w/o reranking	0.135	40.01	0.959	0.786	0.112	41.11	0.954	0.778
	MSMO-Detox	**0.115**	37.42	0.960	0.785	**0.102**	40.23	0.953	0.769
SBF	w/o reranking	0.153	51.12	0.952	0.746	0.149	45.19	0.953	0.756
	MSMO-Detox	**0.131**	47.18	0.954	0.745	**0.133**	42.85	0.956	0.750
DynaHate	w/o reranking	0.209	155.72	0.946	0.720	0.209	125.19	0.944	0.718
	MSMO-Detox	**0.191**	141.01	0.947	0.721	**0.192**	114.35	0.945	0.716
Jigsaw	w/o reranking	—	—	—	—	0.198	190.74	0.935	0.693

MSMO-Detox	—	—	—	—	**0.187**	188.60	0.936	0.692

消融实验结果表明, 在所有数据集上, 加入多目标重排序组件对模型性能的提升至关重要, 能够起到将毒性降低并有效保持文本质量的作用.

4 结论与展望

本文针对现有文本去毒方法忽视隐式毒性与去毒后的文本质量难以得到保障的问题, 提出一种面向隐式毒性的多阶段多目标文本去毒方法. 使用层级解释方法 DecompX, 突出了每个词元对于毒性的具体贡献, 为掩码过程提供有效支撑; 使用 MaRCo 方法进行文本填充, 保证了生成文本有效且无毒; 提出多目标重排序模块, 有效结合 ToxiGen-HateBERT 模块与文本质量评分模块, 使得在有效去毒的同时能够较好地保持文本质量. 然而本文的方法有一定的局限性, 在跨语言场景下的泛化能力还需要进一步验证, 未来的任务将扩展到多语言文本去毒当中.

参考文献

1 于洋, 黄珊, 刘招龙. 新时代网络生态治理策略研究. 中共石家庄市委党校学报, 2023, 25(1): 37–40.

2 干镕华, 高布权. 我国网络生态异化现象及其治理策略. 新媒体研究, 2024, 10(6): 25–29.

3 李东坡, 李媛媛. 论网络社会心态的现代治理. 思想理论教育, 2024(11): 94–99.

4 Dementieva D, Babakov N, Panchenko A. MultiParaDetox: Extending text detoxification with parallel data to new languages. Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. Mexico City: Association for Computational Linguistics, 2024. 124–140.

5 Moskovskiy D, Sushko N, Pletenev S, *et al*. SynthDetoxM: Modern LLMs are few-shot parallel detoxification data annotators. Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies. Albuquerque: Association for Computational Linguistics, 2025. 5714–5733.

6 Bhan M, Vittaut JN, Achache N, *et al*. Mitigating text toxicity with counterfactual generation. Proceedings of the 3rd World Conference on Explainable Artificial Intelligence. Istanbul: Springer, 2026. 135–157.

7 Dementieva D, Babakov N, Ronen A, *et al*. Multilingual and explainable text detoxification with parallel corpora. Proceedings of the 31st International Conference on Computational Linguistics. Abu Dhabi: Association for Computational Linguistics, 2025. 7998–8025.

8 Pour MMA, Farinneya P, Bharadwaj M, *et al*. COUNT: Contrastive unlikelihood text style transfer for text detoxification. Findings of the Association for Computational Linguistics: EMNLP 2023. Singapore: Association for Computational Linguistics, 2023. 8658–8666.

9 Hartvigsen T, Gabriel S, Palangi H, *et al*. ToxiGen: A large-scale machine-generated dataset for adversarial and implicit hate speech detection. Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics. Dublin: Association for Computational Linguistics, 2022. 3309–3326.

10 Mukherjee S, Lango M, Kasner Z, *et al*. A survey of text style transfer: Applications and ethical implications. arXiv:2407.16737, 2024.

11 Dale D, Voronov A, Dementieva D, *et al*. Text detoxification using large pre-trained neural models. Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing. Association for Computational Linguistics, 2021. 7979–7996.

12 Feng YX, Yi XY, Wang XT, *et al*. DuNST: Dual noisy self training for semi-supervised controllable text generation. Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics. Toronto: Association for Computational Linguistics, 2023. 8760–8785.

13 Lee B, Kim H, Kim K, *et al*. XDetox: Text detoxification with token-level toxicity explanations. Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing. Miami: Association for Computational Linguistics, 2024. 15215–15226.

14 Floto G, Pour MMA, Farinneya P, *et al*. DiffuDetox: A mixed diffusion model for text detoxification. Findings of the Association for Computational Linguistics: ACL 2023. Toronto: Association for Computational Linguistics, 2023. 7566–7574.

15 Guo YJ, Cui GQ, Yuan LF, *et al*. Controllable preference optimization: Toward controllable multi-objective alignment. Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing. Miami: Association for Computational Linguistics, 2024. 1437–1454.

16 Liu Y, Liu XY, Zhu XR, *et al*. Multi-aspect controllable text generation with disentangled counterfactual augmentation. Proceedings of the 62nd Annual Meeting of the Association

for Computational Linguistics. Bangkok: Association for Computational Linguistics, 2024. 9231–9253.

17 Pesaranghader A, Verma N, Bharadwaj M. GPT-DETOX: An in-context learning-based paraphraser for text detoxification. Proceedings of the 2023 International Conference on Machine Learning and Applications. Jacksonville: IEEE, 2023. 1528–1534.

18 Bevendorff J, Dementieva D, Fröbe M, *et al*. Overview of PAN 2025: Generative AI detection, multilingual text detoxification, multi-author writing style analysis, and generative plagiarism detection. Proceedings of the 47th European Conference on Information Retrieval. Lucca: Springer, 2025. 434–441.

19 Řehulka E, Šuppa M. RAG meets detox: Enhancing text detoxification using open large language models with retrieval augmented generation. Proceedings of the 2024 Conference and Labs of the Evaluation Forum. Grenoble: CEUR Workshop Proceedings, 2024. 3021–3031.

20 Hallinan S, Liu A, Choi Y, *et al*. Detoxifying text with MaRCo: Controllable revision with experts and anti-experts. Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics. Toronto: Association for Computational Linguistics, 2022. 228–242.

21 Modarressi A, Fayyaz M, Aghazadeh E, *et al*. DecompX: Explaining Transformers decisions by propagating token decomposition. Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics. Toronto: Association for Computational Linguistics, 2023. 2649–2664.

22 Sap M, Gabriel S, Qin LH, *et al*. Social bias frames: Reasoning about social and power implications of language. Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics. Association for Computational Linguistics, 2020. 5477–5490.

23 Vidgen B, Thrush T, Waseem Z, *et al*. Learning from the worst: Dynamically generated datasets to improve online hate detection. Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing. Association for Computational Linguistics, 2021. 1667–1682.

(校对责编: 张重毅)