From Toxicity to Constructive Dialogue: An LLM-Driven Detoxification Approach with Multi-Source Parallel Data

by

Md Fardin Parvez
23341099
Aswadul Karim Rashik
24341250
MD Yameem Daiyan Deepto
21241052
Mostakim Ul Haq
23241080
Alex Noor Khan
18241005

A thesis submitted to the Department of Computer Science and Engineering
in partial fulfillment of the requirements for the degree of
B.Sc. in Computer Science

Department of Computer Science and Engineering
Brac University
June 2025

Declaration

It is hereby declared that

1. The thesis submitted is my/our own original work while completing degree at Brac University.
2. The thesis does not contain material previously published or written by a third party, except where this is appropriately cited through full and accurate referencing.
3. The thesis does not contain material which has been accepted, or submitted, for any other degree or diploma at a university or other institution.
4. We have acknowledged all main sources of help.

Student's Full Name & Signature:

Md Fardin Parvez
23341099

Aswadul Karim Rashik
24341250

MD Yameem Daiyan Deepto
21241052

Mostakim Ul Haq
23241080

Alex Noor Khan
18241005

Approval

The thesis/project titled "From Toxicity to Constructive Dialogue: An LLM-Driven Detoxification Approach with Multi-Source Parallel Data" submitted by

1. Md Fardin Parvez (23341099)
2. Aswadul Karim Rashik (24341250)
3. MD Yameem Daiyan Deepto (21241052)
4. Mostakim Ul Haq (23241080)
5. Alex Noor Khan (18241005)

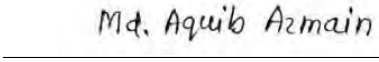
Of Spring, 2025 has been accepted as satisfactory in partial fulfillment of the requirement for the degree of B.Sc. in Computer Science on June, 2025.

Examining Committee:

Supervisor:
(Member)

MD. Fakhruddin Gazzali
Lecturer
Department of Computer Science and Engineering
Brac University

Co-Supervisor:
(Member)



Md. Aquib Azmain
Lecturer
Department of Computer Science and Engineering
Brac University

Co-Supervisor:
(Member)

Dr. Farig Yousuf Sadeque
Associate Professor
Department of Computer Science and Engineering
Brac University

Head of Department:
(Chair)

Sadia Hamid Kazi, PhD
Chairperson and Associate Professor
Department of Computer Science and Engineering
Brac University

Abstract

The surge of toxic interactions online poses a serious obstacle to cultivating constructive digital spaces. Current automated detoxification systems, though encouraging in theory, often stumble: they veer off-topic (semantic drift), strip away too much nuance (over-detoxification), and operate like black boxes, eroding trust and complicating real-world use. In response, this paper outlines a detoxification pipeline designed to tackle these issues head-on by weaving explicit reasoning into grounded generation. Our proposed pipeline particularly introduces a multi-task setup that fine-tunes a 7-billion-parameter language model in two stages: first, it produces a clear explanation of why a comment is toxic (drawing on the concept of Chain-of-Thought prompting), and then it rewrites the comment. What sets our approach apart is a three-step inference routine: the explanation guides a retrieval query that fetches relevant, benign examples from a curated knowledge base via Retrieval-Augmented Generation. The final rewrite leans on both the rationale and these examples, anchoring the output in a transparent reasoning path and verified data. To bolster robustness, we merge data from over 40,000 cases across three sources—synthetic examples with reasoning (DetoxLLM), human-crafted paraphrases (ParaDetox), and adversarial hate speech instances (ToxiGen)—using a fresh fusion strategy. Crucially, we make the fine-tuning doable on everyday hardware by employing 4-bit Quantized Low-Rank Adaptation (QLoRA). Experiments show that our reasoning-plus-retrieval pipeline is a feasible and effective alternative at retaining the original meaning while achieving strong style-transfer metrics. By emphasising not only performance but also clarity and fidelity, this system charts a path toward more accountable, inspectable moderation tools.

Keywords: Detoxification; Text Detoxification; Toxicity; Chain-of-Thought Reasoning (CoT); Retrieval-Augmented Generation (RAG); Semantic Drift; Bias Mitigation; QLoRA (Quantized Low-Rank Adaptation); LLM; Multi-Source Parallel Data; Hybrid Datasets; Cross-Platform Generalization; Robustness in Language Models

Acknowledgement

To begin, we sincerely express our thankfulness and praise to Allah, whose divine direction has enabled us to conclude our thesis without severe setbacks. Second, we would like to thank our supervisor, MD. Fakhruddin Gazzali, as well as our co-supervisors, Md. Aquib Azmain and Dr. Farig Yousuf Sadeque, for their continuous support and essential guidance during our research journey. They have had an important role in developing our work. Finally, we would like to express our heartfelt gratitude to our parents for their constant support and prayers, which have been the foundation of our successes, bringing us to the edge of graduation. The completion of our thesis has been a journey filled with heavenly blessings, mentoring, and constant family support. We recognise and regard each of these factors as essential to our success, and we look forward to using the lessons and principles we have learned from this experience in our future endeavours with great humility and appreciation.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 The Gravity of Online Toxicity and the Need for Advanced Mitigation

Online platforms, from social media networks such as Facebook, Twitter and Reddit to collaborative environments like Wikipedia and GitHub serve as vital spaces for interactions, knowledge sharing, and discussions. However, features such as anonymity, scale, and speed that enable this type of global exchange also contribute to fostering environments where toxic communication can thrive. This toxicity encompasses explicit hate speech and personal attacks, as well as subtle microaggressions, sarcasm and condescension [7]. Such types of interactions pose a critical threat to the sustainability of online communities causing users to alienate themselves which results in diminishing productivity. The prevalence of toxic behaviour is not a niche issue but a widespread societal problem. Quite surprisingly, even with its collaborative ethos, Wikipedia is not immune to toxic conduct either. A study by Wulczyn et al. [4], where 63 million revisions were examined, revealed that 1.2% of edits contained personal attacks where newcomers were disproportionately targeted, discouraging participation altogether. The number of personal attacks in those revisions amounted to approximately 1 million which is a significantly high number of attacks given the apparent straightforward informative nature of the platform. This large-scale analysis of Wikipedia discussion comments revealed that personal attacks are not sporadic but rather quite persistent and frequent. Additionally, vandalism—often toxic in nature—affected many edits, calling for automated detection methods [1]. These studies show that even ostensibly neutral websites such as Wikipedia find it difficult to keep toxicity at bay.

The problem extends to professional collaborative platforms such as GitHub as well. For example, 21% of GitHub contributors report experiencing toxic interactions, often leading to disengagement from projects [28]. Founta et al. [5] note that toxicity's pervasiveness leads to consequences that are particularly severe in adolescents engaging in platforms like X (formerly Twitter), contributing to the degradation of mental health. This problem is not static; it is accelerating. The rise in online hostility is inseparably linked to tangible, offline harm. Sociological and criminological research has established a disturbing connection between online hate speech and subsequent hate crimes [36]. A particularly woeful example would be the overwhelmingly toxic

anti-Asian rhetoric online during the COVID-19 pandemic. A 77% increase was observed in reported hate crimes against the Asian American and Pacific Islander (AAPI) community in the United States in 2020 [36]. This pattern is reactive, with major geopolitical events triggering immediate and sharp increases in online hate. Beyond individual or group-directed harm, the constant exposure to such language encourages a broader societal "desensitization". As a consequence, there is an entire process where empathy erodes and prejudice is normalized, creating a fertile ground for radicalization. According to Chandrasekharan et al. [2], postings in unmoderated communities on Reddit violate platform standards in various capacities, and toxic content there spreads disproportionately due to the high prevalence. Davidson et al. [3], found that linguistic signs of hate speech were present in 5% of sampled messages on Twitter even though their research underlines the challenges faced in order to distinguish between hate speech from offensive language and noted that certain types of hate speech, such as racist tweets had a higher likelihood to be classified as hate speech as opposed to sexist tweets which were more frequently classified as offensive language. These results demonstrate how toxicity is made worse by platform design and moderation flaws, creating hostile environments that deter constructive discourse. Therefore, the serious societal cost of online toxicity thus establishes an urgent need for effective, scalable and responsible intervention systems or mechanisms.

1.2 Limitations of Existing Detoxification Approaches

In response to this challenge, online platforms have relied on a combination of manual and automated content moderation. Human moderation, while capable of exercising subtle judgement through a refined understanding of the particular platforms' nuances, is expensive and psychologically taxing for moderators. This renders manual content moderation unfeasible and unscalable where millions of users are present. Consequently, automated systems have been given the precedence and thus adopted more widely. In the early stages, automated approaches primarily focused on toxicity detection. These systems often implemented keyword-based filters or simple machine learning classifiers. However, these methods have proven to be insufficient and not robust. Keyword-based systems are easily bypassed through adversarial techniques such as character-level obfuscation (e.g., "sh!t" to evade filters) where toxic words are intentionally altered to avoid detection. Researchers have addressed this with character-level encoders, though these methods demand more computational resources. Racial and gender biases in current detectors, trigger false positives or negatives and pose another significant challenge, especially in technical reviews and comments where nuance is often missed. In contrast, even though standard classifiers are able to perform more sophisticatedly in textual tasks, they often lack the contextual understanding to differentiate between genuine hate speech and nuanced discussions of sensitive topics.

Davidson et al. [3] state that these tools are prone to systemic biases too causing frequent misclassification of non-toxic content from minority communities as offensive. Toxic messages are often camouflaged in sarcasm, ambiguous technical terms, or typos, which make their detection complex. This means that effective mitigation

of toxicity calls for context-sensitive tools. Furthermore, Gildaa et al. [16] aptly points out a key detail that the NLP community has done extensive research on more direct forms of toxicity and abuse and has given less attention to the more subtle forms of abuse under the title "unhealthy conversations". Previous studies focused on overly toxic, violent and insulting comments, but the dataset they used focuses on the subtle forms of abuse that inflict pain on others without being explicitly abusive. State-of-the-art (SoTA) detection tools, such as Google's Perspective API also faced similar challenges. They struggle with implicit or context-dependent toxicity. Recognising the limitations of the aforementioned simple detection was crucial since it has motivated a fundamental change towards a more constructive intervention: text detoxification. In the field of Natural Language Processing (NLP), detoxification is classified as a text style transfer (TST) task. Detoxification aims to flag or delete harmful messages and to rewrite them into non-toxic equivalents while preserving the original author's core intent [27]. This method offers the promise of rehabilitating conversations and educating users, rather than simply censoring them. Text Detoxification using Large Pre-trained Neural Models Dale et al. [15] notes that automatic rewriting of offensive content, or text detoxification, is an active but less explored area of research. Detoxification requires better preservation of original meaning than many other style transfer tasks, such as sentiment transfer, and therefore should be approached differently.

Despite substantial progress, the current methods are plagued with limitations that impede their practical deployment. (1) First, "hallucination" and semantic drift remain fundamental challenges to tackle in the process of detoxification. It is a difficult task for the models to preserve the core message of the original text, and many models fail in this particular aspect. They produce a fluent, non-toxic sentence but the meaning is entirely different. Although LLMs such as GPT-2, fine-tuned with prompt-based training, are able to rewrite toxic content into more neutral phrases, characteristic meaning is still a challenge to maintain, since models such as BART-Detox struggle to maintain consistency and GPT-3.5 sometimes introduces contextual shifts unintended by the user, further complicating how semantics are preserved during detoxification [24]. This is a critical failure, as it misinterprets the user's intent and can be more damaging than no moderation at all. (2) Second, the vast majority of detoxification models operate as "black boxes" and hence lack transparency and interpretability. Although they provide a rewritten output, no justification or explanation is included for the change [34]. This kind of opacity makes it difficult for users to understand the reasoning behind the change. It is not just difficult for the users but also for auditors and developers. The auditors may struggle to verify compliance with moderation policies, and developers would be unable to debug failure cases. The lack of transparency emerging from the absence of explanations is a significant barrier to building trust and ensuring responsible system behaviour. (3) Third, existing models frequently suffer from poor domain generalization which in turn indicates a lack of cross-platform corpora-trained models. A model that is trained on the linguistic conventions and style of only one platform (e.g., X/Twitter) is probably to perform poorly on another (e.g., Reddit), as community norms, and toxic expressions vary significantly [24]. (4) Fourth, there is a lack of a safety gate due to which models are completely incapable of handling "non-detoxifiable" content. Non-detoxifiable messages are those in which the toxicity is

so intertwined with the core meaning that any kind of rewrite would fundamentally change it. Almost no system, except only one, DetoxLLM by Khondaker et al. [34], have a mechanism to recognise these cases and may produce nonsensical outputs instead of flagging the input as unworkable or even issue a warning regarding the output being semantically incoherent.

Overall, this kind of fragility necessitates models that can be trained on diverse, multi-domain/cross-platform parallel data to effectively detoxify texts while preserving its original meaning and achieve the required robustness for real-world applications. The junction of these limitations reveals that a more intricate and refined system or framework is needed whereby the models do not merely restrict themselves to simple input-output mapping but rather move towards a more reasoned and grounded generation process.

1.3 Thesis Contributions

The central limitations of current detoxification systems such as semantic drift, opacity, poor generalization and inability to handle non-detoxifiable cases stem from a common root: an unconstrained generation process that lacks explicit reasoning, grounding and safety checks of whether the original meaning is kept. It is posited through this research that by enforcing steps where the model is bound to first articulate why a text is toxic and then ground its rewrite in verified examples of non-toxic comments, these fundamental flaws can be systematically addressed. The thesis, therefore, introduces and evaluates a framework, known as ReRight, for text detoxification that integrates Chain-of-Thought (CoT) rationale generation with Retrieval-Augmented Generation (RAG) to produce rewrites that are not only less toxic but also more semantically faithful, interpretable, and safer. By fine-tuning a large language model (LLM) on a fused corpus of synthetic, human-crafted, adversarial data, and using the memory-efficient QLoRA technique, we demonstrate that this CoT+RAG approach systematically addresses the key limitations of semantic drift and opacity which were prevalent in previous methods. This approach establishes a new system for grounded and explainable detoxification.

This research makes the following five principal contributions:

1. **Data Fusion for Robustness**: We pioneer the amalgamation of three distinct datasets—DetoxLLM (synthetic with CoT), ParaDetox (human-paraphrased), and ToxiGen (adversarial)—to construct a fine-tuning corpus that confers robustness against a wide spectrum of toxic phenomena.

2. **Multi-Task Rationale Supervision**: We propose a multi-task learning objective that jointly trains the model to generate both an explanatory rationale and the final detoxified output, explicitly architecting the system for interpretability.

3. **The CoT+RAG Grounded Generation Pipeline**: We design and implement a novel three-step inference pipeline that first generates a rationale

(CoT), then uses it to retrieve relevant examples (RAG), and finally generates a rewrite grounded in both the rationale and the retrieved context.

4. **Domain Conditioning for Style Adaptability**: We incorporate a data_source token during training, explicitly conditioning the model to adapt its generation style to different online domains (e.g., Reddit, Twitter), thereby improving its generalization capabilities.

5. **Efficient Fine-Tuning on Consumer Hardware**: We provide a practical demonstration of how Quantized Low-Rank Adaptation (QLoRA) enables the fine-tuning of a complex, multi-task, multi-dataset model on a single consumer-grade GPU, making this advanced research methodology accessible to the broader academic community.

1.4 Report Structure

This report is structured as follows. **Chapter 2** reviews the relevant literature on toxicity, text style transfer, and the foundational methodologies used in this work. **Chapter 3** presents a detailed description of our proposed CoT+RAG methodology, including the data fusion strategy, model architecture, and inference pipeline. **Chapter 4** outlines the experimental setup, defining the datasets, evaluation metrics, and baselines. **Chapter 5** presents and analyzes the quantitative and qualitative results of our experiments. **Chapter 6** discusses the implications, practical challenges, and limitations of this study. Finally, **Chapter 7** concludes the report and suggests promising directions for future research.

Chapter 2

Related Work

2.1 Defining and Detecting Online Toxicity

Toxicity is a prevalent issue that influences various scenarios from interpersonal relationships to professional interaction, especially where communication and collaboration are crucial. Toxic communication can be defined as the kind of communication that hinders social norms with a primary focus on destroying a social contract that includes the use of offensive language, mockery, verbal abuse in addition to arguable aggressive criticism [14]. In addition to that, it could also be extended to any kind of speech or text that can be disrespectful, hostile, or that exhibits offensive speech to hurt or provoke people. Toxic language takes the forms of insults, personal attacks, hate speech, microaggressions, trolling, and condescension—all in their forms and intensities [7]. Toxic communication can range from explicit hate speech and harassment to subtle forms of negativity such as trolling and passive-aggressive comments [14]. In the paper Predicting Different Types of Subtle Toxicity in Unhealthy Online Conversations, the classification of toxicity used in their dataset is unique and granular [16]. The subtle abuse types are captured, coding each comment with one or more of seven specific forms of subtle toxicity: antagonize, condescending, dismissive, generalization, generalization unfair, hostile, and sarcastic, alongside healthy comments. Toxic behaviour is not always random; sometimes it can stem from the structural design of online platforms. Most often certain algorithms give precedence to sensitive or disruptive information and this ultimately results in negative engagement [10]. For toxicity detection, Kumar et al. [35] prompted LLMs with a definition of toxic content: "a rude, disrespectful, or unreasonable comment that is likely to make someone leave a discussion". Toxicity can be at the same time context-dependent, where factors like cultural norms, geography, and medium of communication influence how it is perceived and handled. It can range from anything that can make participants leave a discussion to language containing vile terminology, swearing, racism, or anything that is socially, culturally, morally, or religiously offensive [28].

Toxic interactions in online communities—whether explicit hate speech, subtle incivility, or adversarial behaviors—undermine collaboration, reduce participation, and harm user well-being. Studies consistently demonstrate that toxicity hinders knowledge-sharing, repels contributors, and delays progress on open-source software-

engineering (SE) platforms and technical forums [20][2]. As mentioned previously, it is incumbent to understand the severity of harbouring toxicity in any online setting or platform. Wulczyn et al. [4] discovered, for instance, that newcomers are disproportionately the targets of personal attacks on Wikipedia, which raises editor attrition rates [4]. Similarly, according to GitHub surveys, 50% of developers have encountered toxic interactions, and 21% of developers have considered completely quitting projects due to hostility [13][22]. These results demonstrate how toxicity undermines trust, which is an essential component of productive collaboration.

These challenges are further flamed due to the physiological factor of toxicity. Users under such toxicity face more stress, poor job satisfaction along with burnout. This is especially the case for voluntary workforces [12]. For example, there is a direct correlation between reduced editor engagement and uncivil language as people naturally try to avoid conflict [7]. Reddit, for example, shows that most unmoderated threads lack intelligent and productive discussions as users avoid debates [2]. Such threads usually inflict greater harm to marginalised communities, exacerbating the lack of participation [17].

We must also consider practical expenses. Manual moderation systems utilized by Stack Overflow struggle to process high bandwidth of toxic content and as such, many of these are either delayed or straightened to avoid the inspection. Egelman et al. [9] notes that subjective interpretations of toxicity—where blunt feedback is deemed constructive in some cultures and hostile in others—complicate moderation. This bias is due to differences in language: sarcasm, coded language, and platform-specific slang (e.g., Reddit's "/s" indicator) evade rule-based detection systems, allowing toxic interactions to persist [33]. Consequently, communities face slower project timelines, wasted resources, and fragmented communication, undermining their core missions.

Sometimes the subject nature of the toxic behavior makes it more challenging to regulate, meaning that one person perceives as constructive criticism what another person may consider toxic. This subjectivity is largely influenced by factors like geographical location, cultural background, and the hierarchical position within the community [5]. For instance, direct and blunt communication might be more accepted in some SE cultures, while others might view the same behavior as overly aggressive or toxic [10]. Toxic behavior may reduce productivity, increase stress, and even trigger developer burnout in SE communities where teamwork and mutual respect are essential for success. Toxic team members may engage in manipulation, exclusion, or belittling comments to assert power or dominance.

The task of automatically detecting such content has evolved significantly. As mentioned previously, early systems relied on simple keyword blocklists and rule-based filters. These methods were not robust and thus were easy to evade. This is due to their lack of semantic understanding of the diverse nature of the toxic language [39]. As stated previously, machine learning's rise ushered in classifiers of increasing sophistication, trained meticulously on annotated datasets. Today, large Transformer-based architectures—BERT and its successors—have been at the forefront, consistently demonstrating impressive accuracy in detecting toxic content [6]. Google's

Perspective API, the most publicly used toxicity classifier, assigns a toxicity score to any text snippet, is built on these models and have become staples in both research and industry. However, even these SoTA classifiers fall short. A significant body of research conducted by Hartsvigsen et al. [18] has revealed they often absorb and even magnify the biases lurking in their training data. Take, for example, the troubling tendency to link mentions of minority identities with toxicity: a benign remark about a marginalized group can be mislabeled as harmful. Such pitfalls remind us that mere detection is not enough; instead, we need richer strategies like detoxification approaches that revise problematic content rather than simply silencing it.

2.2 Paradigms in Automated Text Style Transfer for Detoxification

A Taxonomy of Detoxification Methods

The approaches to text detoxification have evolved significantly over time. A structured taxonomy helps to understand this progression and the trade-offs inherent in each paradigm. The following table summarizes the key methods.

Method	Key Idea	Representative Work(s)	Strengths	Limitations
Delete-Retrieve-Generate	Delete toxic spans, retrieve non-toxic fragments, and generate a new sentence around them.	Gandhe& Logacheva (2021)[19]	Simple, interpretable pipeline.	Static retrieval can break fluency; no explicit reasoning; often grammatically awkward.
Mask-and-Fill Paraphrasing	Mask toxic words identified by a classifier and use a language model to fill in the blanks.	Dale et al. (2021)[15]; Wu et al. (2019b)[4]	Fully unsupervised; no parallel data needed.	Can over-mask, hallucinate, and produce content-unaware edits; struggles with multi-word replacements.
Disentanglement-based Models	Learn separate latent representations for style and content; transfer style by manipulating the style vector.	Shen et al. (2017)[38]	Theoretically elegant; allows explicit modeling of style and content.	Achieving perfect disentanglement is exceptionally difficult; often leads to degraded content or incomplete style transfer.

Continued on next page

Table 2.1 – continued from previous page

Method	Key Idea	Representative Work(s)	Strengths	Limitations
Supervised Fine-Tuning	Fine-tune a pretrained sequence-to-sequence model (e.g., BART, T5) on a parallel corpus of toxic/non-toxic pairs.	Lewis et al. (2019)[8]; Raffel et al. (2020)[11]; Logacheva et al. (2022)[19]	Generally achieves the best performance (SOTA); leverages power of pretrained models.	Dependent on high-quality parallel data; can be computationally expensive; susceptible to semantic drift and lack of transparency.
Reinforcement Learning (RL)	Use a reward model (e.g., toxicity classifier) to directly optimize a generation policy for non-differentiable metrics.	General RL paradigm (e.g., for TST); RLHF (Ouyang et al., 2022)[21]	Can directly optimize for desired metrics like style accuracy; theoretically appealing for complex objectives.	Notoriously difficult and unstable to train; high variance; computationally expensive; reward model is a black-box.
RLHF / Preference Tuning	Use human preferences or a toxicity classifier's feedback to steer generation away from toxic outputs	General RLHF paradigm	Aligns with human judgments.	Expensive (human annotation); black-box reward; no fine-grained explanations.
Controllable Generation	Use techniques like adding a loss that penalizes generating known toxic tokens.	Zhang et al., (2023)[40] (Unlikelihood)	Direct token-level control; lightweight compared to full RL.	Does not teach why toxicity occurs; can lead to overly generic outputs; simple unlikelihood can penalize non-toxic content.
LLM Prompting (Zero/Few-Shot)	Instruct a large, general-purpose LLM (e.g., GPT-4) to perform detoxification without fine-tuning.	GPT-DETOX (Pour et al., 2024)[27]	No training/fine-tuning required; highly flexible; computationally cheap to set up.	Performance can be unstable; requires careful prompt engineering; can be prone to hallucination; may not match performance of specialized fine-tuned models.

Continued on next page

Table 2.1 – continued from previous page

Method	Key Idea	Representative Work(s)	Strengths	Limitations
Activation Engineering / Steering	Modify a model's internal activations at inference time with a "steering vector" to guide output towards non-toxicity.	Turner et al., (2023)[29]; Zou et al., (2023)[30]	Fine-tuning-free; lightweight and computationally cheap; offers fine-grained, composable control.	Can be unreliable if behavior is not linearly represented; high steering strength can degrade text quality.
Explainability-Driven Detoxification (This method has been leveraged)	Integrate token-level explanations (XAI) to precisely identify toxic spans for more accurate masking and infilling.	DetoxLLM (Khondaker et al., 2024)[34]	Improves precision of edits, reducing over/under-masking; enhances transparency and trustworthiness.	Performance is dependent on the quality of the underlying explanation method; it adds a step to the pipeline.

Table 2.1: Comparison of Detoxification Methods

These earlier methods laid important groundwork but often struggled with the delicate balance of removing toxicity while preserving meaning and fluency. The advent of LLMs has enabled more powerful and flexible approaches.

2.3 The Emergence of Large Language Models in Content Moderation

2.3.1 Open-Source LLMs in Text Processing

Toxicity detection tools are vital for flagging harmful comments, however it does not offer any proactive solution to tackle online abusive interactions. When combined with the detoxification process, these tools can exchange positive interactions in the open society. In this case, transformer models like BERT, GPT-2, and GPT 3.5 are widely used.

BERT is a transformer model used in Natural Language Understanding (NLU) that understands a context by its bidirectional attention mechanism. As it can capture the nuances of both past and future words, it flourishes in toxicity detection thus making it ideal for understanding the intent and sentiment of a comment. On the other hand, Agarwal et al. [24] showed Large Language Models (LLM) perform well

on this hate rephrasing task, outperforming state-of-the-art baselines like BART-Detox. The reason why LLMs work well is the fact that they can effectively perform the three underlying sub-tasks of hate speech rephrasing: hate speech detection, hate span identification, and then rephrasing the hateful spans. The paper also argues that GPT-3.5 particularly excels, not only outperforming baseline and open-source models but also outperforming human-generated ground-truth rephrasings in the dataset in terms of hate intensity reduction, relevance, and hallucination scores. Moreover, the models were fine-tuned with prompts to detect and rephrase only text spans with hate speech without changing the meaning of the original text. Their results also showed that GPT-3.5 can sometimes generate considerably different contextual meanings and extra information from the web, based on the task descriptions and definitions provided in prompts. The algorithm in A Toxic Style Transfer Method Based on the Delete–Retrieve–Generate Framework Exploiting Toxic Lexicon Semantic Similarity, Iglesias et al. [26] introduces a parameter K, which averages the top-K similarities of a word with a lexicon. This helps alleviate issues where common words might be mistakenly deleted due to high maximum embedding similarity with a single toxic word (e.g., "portrayed" and "betrayed"). This ensures that only genuinely toxic words, which have consistently high average similarities across a lexicon, are targeted for removal.

Dale et al. [15] in their paper utilised CondBERT. CondBERT enforces the generation of non-toxic words by calculating the toxicity of each token in BERT's vocabulary using an external classifier and penalizing the predicted probabilities of tokens with positive toxicities. This control mechanism ensures high style accuracy. Furthermore, it incorporates specific heuristics to preserve content, such as preserving original tokens instead of masking them and reranking replacement words based on their semantic similarity to the original words.

Watch Your Language: Investigating Content Moderation with Large Language Models Kumar et al. [35] argues that LLMs, specifically GPT-3.5, can be effective at rule-based community moderation. They also argue that modern LLMs (GPT-3, GPT-3.5, GPT-4, Gemini Pro, LLAMA 2) significantly outperform widely used toxicity classifiers like Google Jigsaw's Perspective API as GPT-3.5 achieved an accuracy of 0.73 and an F1 score of 0.75, showing the most balanced performance. Another key argument in the paper is that despite increases in LLM model size, there is only marginal improvement in toxicity detection performance. The study observed similar F1 scores across GPT-3, GPT-3.5, and GPT-4, showing a performance limit for LLMs on this task. This implies that simply increasing model size may not lead to significant further benefits in content moderation. It also argues that providing a clear definition of toxicity is crucial for performance, as omitting it can significantly reduce recall.

While dealing with implicit toxicity, Toxigen presented by Hartvigsen et al. [18], came to the picture which is designed to improve the implicit and adversarial hate speech. The original dataset contains 274,186 sentences that are distributed across 13 minority groups of which 98.2% are toxic. Toxigen works with ALICE abbreviated as Adversarial Language Imitation with Constrained Exemplars, an algorithm that produces adversarial examples by pitting an LLM like GPT-3 against a toxicity

classifier. It has been observed that when toxicity classifiers are fine-tuned, ToxiGen provides a 7-19% improvement on datasets like ImplicitHateCorpus and SocialBias-Frames, suggesting improvement in generalization. But it does face some challenges, especially during the implicit hate speeches as they are often can lead to ambiguity. While Large Language Models (LLMs) show promise in content moderation, Kumar et al. [35] argues that they are not perfect and still require human review and careful guardrails before widespread adoption. LLMs often miss implicit toxicity and may struggle with decisions requiring significant context about what is being said.

Further explorations into several LLM-driven frameworks and pipelines were necessary to address the challenge of generalization across different platforms and also for semantic coherence. One such framework is DetoxLLM, proposed by Khondaker et al. [34] which is a detoxification framework that was developed to rewrite toxic language into non-toxic alternatives while also maintaining semantic coherence. Unlike other approaches, which utilize an in-platform detection system, DetoxLLM uses a cross-platform pseudo-parallel dataset that incorporates cross-platform generalization. It is the first cross-platform detoxification framework that generalizes across multiple platforms and is more robust and effective in capturing linguistic variations in toxic language across different platforms including Wikipedia, Reddit, Facebook, YouTube, and Twitter. DetoxLLM also uses a module to generate explanations to justify why a particular text or comment is labeled as toxic. This promotes healthier communication among users by increasing transparency and credibility. Following this explanation generation module, it has a paraphrase detector for situations that cannot be detoxified-for example, rephrasing a particular toxic text can sometimes completely change the meaning of the comment made. Furthermore, in the detection of adversarial attacks, DetoxLLM is ahead of the rest. The DetoxLLM outperformed all state-of-the-art detoxification models during benchmark tests, especially cross-platform. DetoxLLM (LLaMA-CE) achieved 97.21% accuracy on adversarial token detection, 67.50% high-quality detoxification responses (A B ratings) compared to ParaDetox's 40.63%, 55% lower toxicity in non-detoxifiable cases, and 70% convincing toxicity explanations [34] [19]. However, DetoxLLM heavily relies on ChatGPT-generated pseudo-parallel datasets for training. This kind of synthetic data may lack the linguistic diversity of toxic language used by real people and also introduces biases from the LLM used for data generation.

Contrasting the synthetic unsupervised dataset generated by ChatGPT, used by DetoxLLM, the ParaDetox dataset is a parallel detoxification dataset made using a crowd-sourced human annotated method that significantly improves detoxification accuracy and fluency [19]. The dataset includes 12,000 toxic sentences, each with one to three human-annotated non-toxic paraphrases, and further includes 1,400 toxic-neutral sentence pairs extracted from the ParaNMT dataset, manually filtered for quality and alignment with the detoxification task [19]. Models trained on ParaDetox outperform unsupervised detoxification models by reaching a BLEU score of 64.53, considerably higher than Mask&Infill at 52.47, CondBERT at 42.45, and ParaGeDi at 25.39 [15] [19]. Whereas for higher toxicity reduction scores (STA), the unsupervised models have higher scores: CondBERT with 0.98, and ParaGeDi with 0.99. Again, the model of ParaDetox better preserves the content, having SIM = 0.86 compared to the 0.77 of CondBERT and 0.71 of ParaGeDi, while being more

fluent, FL = 0.89 as compared to the 0.82 of CondBERT and 0.88 of ParaGeDi [15]. This underlines that parallel data can significantly enhance the quality of the detoxified text, outperforming many times the detoxified text produced by purely synthetic data. However, unlike DetoxLLM, the ParaDetox dataset does not contain explanations for why a text is toxic, as the humans involved with the making of the dataset were only involved in rephrasing toxic language, and not explaining them.

2.4 Foundational Methodologies

2.4.1 Chain-of-Thought (CoT) Prompting for Interpretability

The demand for transparency in AI systems is especially acute in contexts where AI moderates human expression. The traditional detoxification models do work, but they fail to provide insight into their decision making process. A model might rephrase a toxic comment but it leaves the user guessing why the original was deemed toxic and how the paraphrasing addresses the specific issues. This acts as a barrier to user trust and system accountability. To tackle this, Chain of Thought (CoT) prompting has emerged as a powerful approach to strengthen the reasoning capabilities of Large Language Models. CoT allows LLMs to break down a complex problem into a series of intermediate logical steps rather than collapsing responses into a single token sequence before making a final conclusion [39]. Wei et al. [23] showed enabling CoT reasoning through complex tasks step by step led to performance improvement as high as 30% accuracy gain on mathematical reasoning benchmarks like GSM8K with models greater than 62B parameters. Latest works extended the usefulness of CoT beyond arithmetic. Zhang et al. [40] with in particular Pattern-CoT revealed selecting examples by reasoning patterns like comparison, sorting etc. than semantic similarity leads to greater improvement across datasets like Coin and Date where operation patterns are not explicitly defined. Even though Chain of Thought (CoT) is not still a mainstream feature for detoxification datasets especially for ParaDetox and Toxigen, its application is slowly emerging in explainable toxic language paraphrasing. The interpretability of CoT is ideally suited for detoxification as it is inherently subjective and context sensitive. For instance, DetoxLLM although not clearly CoT- based highlighted explanation based paraphrasing, generates models to explain why a given text is toxic before rewriting. This particular structure mirrors CoT prompting where a rationale precedes the final verdict.

However, the existing detox corpora like Paradetox and Toxigen do not provide step by step toxic to non toxic alternatives with intermediate reasoning chains. Thus while working with CoT, it should generate rationales synthetically like the DetoxLLM or just rely on sparse supervision. Again, if the explanation generation is not well supervised, CoT steps may hallucinate. DetoxLLM mitigates this by masking rationale loss when it's empty but at the same time it limits model learning in edge cases. There is also a concern in the faithfulness of the generated rationales too. Research has shown that LLMs can generate logical routes that seem reasonable but are ultimately incorrect, leading to a valid conclusion based on flawed reasoning

[36].

2.4.2 Retrieval-Augmented Generation (RAG) for Grounding

One of the fundamental limitations of standalone LLMs is their property to hallucinate; generate fluent and articulate but factually inaccurate and nonsensical information. This occurs because LLMs are probabilistic models trained on static pieces of data, as a result they are not able to retrieve information in real time [31]. Retrieval-Augmented Generation (RAG) was developed directly to address this problem. RAG architectures basically integrate a parametric language model with a non parametric retrieval element allowing the model to examine external documents before output generation [8]. That means, in a typical RAG pipeline, a user's input is used to retrieve relevant documents, which are then connected with the original prompt and fed as context to the LLM, grounding its response in the provided information [32]. The primary advantages of RAG are its capacity to reduce hallucinations and improve generation accuracy based on knowledge intensive tasks [32]. It effectively grounds the output of the model in a reliable external source. Furthermore, the knowledge base can be updated easily without requiring the need for expensive model retraining, making it a flexible and powerful framework.

However, standard or typical RAG systems usually employ the generation phase with a single, static retrieval step. While this approach works effectively for simple and straightforward queries, it breaks down when dealing with complex problems like when the original query may be ambiguous or insufficient to retrieve necessary context. To overcome this, advanced frameworks have been presented that model retrieval and reasoning as an iterative and dynamic process. A particularly relevant advancement is CoRAG (Chain-of-Retrieval Augmented Generation), which explicitly trains an LLM to carry out multiple retrieval steps, dynamically reformulating the query at each stage according to the changing state of the problem [39]. In multi-hop question-answering tasks, where single-step retrieval frequently fails, CoRAG has shown significant performance gains by breaking down a complex query and iteratively gathering information [39]. For instance, when compared to traditional RAG systems, CoRAG-8B improves the exact match (EM) result on the MuSiQue multi-hop QA benchmark by more than +10 points. Even after using a considerably smaller model than closed-source giants like PaLM or GPT-4, the authors also demonstrate improvements of up to +7.6 BLEU and +5.9 ROUGE-L on KILT tasks like T-REx and HotpotQA [39]. This proves how implementation of RAG where outputs are tied to retrieved contexts helps in reducing hallucination. Also as RAG systems can adjust dynamically to new domains, it thus does not require model retraining providing the system parameter efficiency.

While the application of RAG is widely used in answering questions and factual generation, its application in text detoxification is still emerging. The closest precedent is the architecture of DetoxLLM, but it uses example based prompting not the retrieval method. Thus it lacks domain matched detoxification and stylistic grounding. Again, Řehulka and Šuppa [37] aimed to demonstrate the effectiveness

of using LLMs specially Llama 3 for multilingual text detoxification by adjusting prompts and integrating Retrieval Augmented Generation (RAG). Although in the early stage, it was seen that by incorporating RAG, the average performance improved to 0.403 compared to the base zero shot Llama 3 model achieved an average evaluation score of 0.380 [37]. RAG's core strength lies in dynamically populating the prompt with information from external knowledge bases.

2.4.3 Parameter-Efficient Fine-Tuning: From LoRA to QLoRA

Among the various PEFT methods, QLoRA (Quantized Low-Rank Adaptation) has emerged as a particularly effective solution for fine-tuning large models on single, memory-constrained GPUs [25]. One of the core aspects of the QLora is its ability to load the large, frozen base model weights in a 4-bit quantized format. More specifically, it uses a 4-bit NormalFloat (NF4) data type, which is optimal for weights that follow a normal distribution. This particular step reduces the base model's memory by a factor of 4 compared to 16 bit precision [25]. When it is compared to the standard LoRA, QLora adds small, trainable adapter matrices to the transformer model's layers. It is true that standard LoRA is faster to train but it requires significantly more VRAM, as the base model is typically held in 16 bit precision [25]. Again, QLoRA uses paged optimizers to offload optimizer states to CPU RAM when VRAM runs out in order to handle memory spikes during training. Additionally, it uses "double quantization," which quantizes the quantization constants themselves, further compressing the model and saving additional memory [25].

2.5 Gap Analysis and Positioning of this Work

The preceding review of the literature so far reveals a clear research gap at the intersection of detoxification, interpretability, and grounded generation. Although research on text detoxification is ongoing, existing approaches are always constrained by a trade-off between eliminating toxicity and maintaining meaning and fluency, and they almost universally lack transparency [27]. While approaches like CoT and RAG have shown great power for improving LLM reasoning, interpretability, and faithfulness in other domains, their synthesis and application to the particular problem of text detoxification have not yet been thoroughly explored. In particular, neither the use of RAG to base rewrites in a corpus of confirmed non-toxic cases nor the use of CoT to generate an explicit reason as an intermediary step in a detoxification pipeline have been suggested in previous work.

Therefore, this work is perfectly positioned to fill this gap properly. To the best of our knowledge, it is the first study to develop, put into practice, and rigorously assess a single pipeline that combines Retrieval-Augmented Generation (RAG) for grounding and Chain-of-Thought (CoT) for interpretability in the context of text detoxification. The fundamental weaknesses in the prior art are specifically addressed by this innovative synthesis of methodologies. While the RAG component

is intended to provide a scaffold for faithful, fluid rewriting, hence reducing semantic drift, the CoT component is intended to offer transparency and concentrate the model on the particular toxic parts. This methodological approach, which aims to create a model that is successful, interpretable, robust, and generalizable all at once, is made possible by QLoRA's computing efficiency and reinforced by a strategic data combination of synthetic, human, and adversarial data sources.

Chapter 3

Dataset

3.1 Data Fusion and Preparation

A key hypothesis of this study is grounded on combining diverse and complementary datasets to improve model robustness and generalization capabilities. To build a comprehensive detoxification model, we have used three secondary datasets for this work: DetoxLLM, ParaDetox and Toxigen.

1. **DetoxLLM**: The DetoxLLM dataset is foundational to this study, as it is the sole source of the precise rationales needed for Chain-of-Thought supervision. DetoxLLM employs a cross-platform pseudo-parallel dataset that integrates cross-platform generalization, in contrast to other methods that make use of an in-platform detection system [34]. DetoxLLM consists of a dataset with pairs of texts, toxic and detoxified text, with an explanation of why a certain text is considered toxic. DetoxLLM contains 3 splits: training, validation, and testing which comprises a total of 10455 synthetically generated toxic and non-toxic pairs.

2. **ParaDetox**: ParaDetox is a parallel detoxification corpus that comprises a toxic text together with its neutral, human-annotated version [19]. The dataset consists of 11,939 toxic sentences, with an average of 1.66 non-toxic paraphrases written by human annotators per sentence, which amounts to 19,766 paraphrases [19]. However, only 1 parallel non-toxic alternative will be used for each toxic version to maintain uniformity between the two types of dataset. The purpose of including ParaDetox is to help the model's generations be grounded in fluid, natural-sounding human language and to enhance its capacity to retain semantic content as human paraphrases are generally of higher fidelity than the synthetic ones.

3. **Toxigen**: Toxigen is a large scale. machine generated dataset developed by Hartvigsen et al. [18] containing toxic and non-toxic comments regarding demographic and identity-based groups. It is designed to help train models that can detect subtle and overt hate speech, particularly across a broad range of identity groups. The original dataset contains 274,186 sentences that are distributed across 13 minority groups of which 98.2% are toxic [18]. Each entry in the dataset is focused around an identity group, such as race, religion,

gender, sexuality, nationality, or disability, making it a crucial tool for fairness-aware NLP systems. Each data instance includes a text field (the sentence), a toxicity label (0 for non-toxic, 1 for toxic), and the target identity group it references. ToxiGen is appropriate for training models that need to identify subtle toxicity because the toxic examples vary from explicit hate speech to more covert forms of bias or harmful stereotypes.

Chapter 4

Methodology

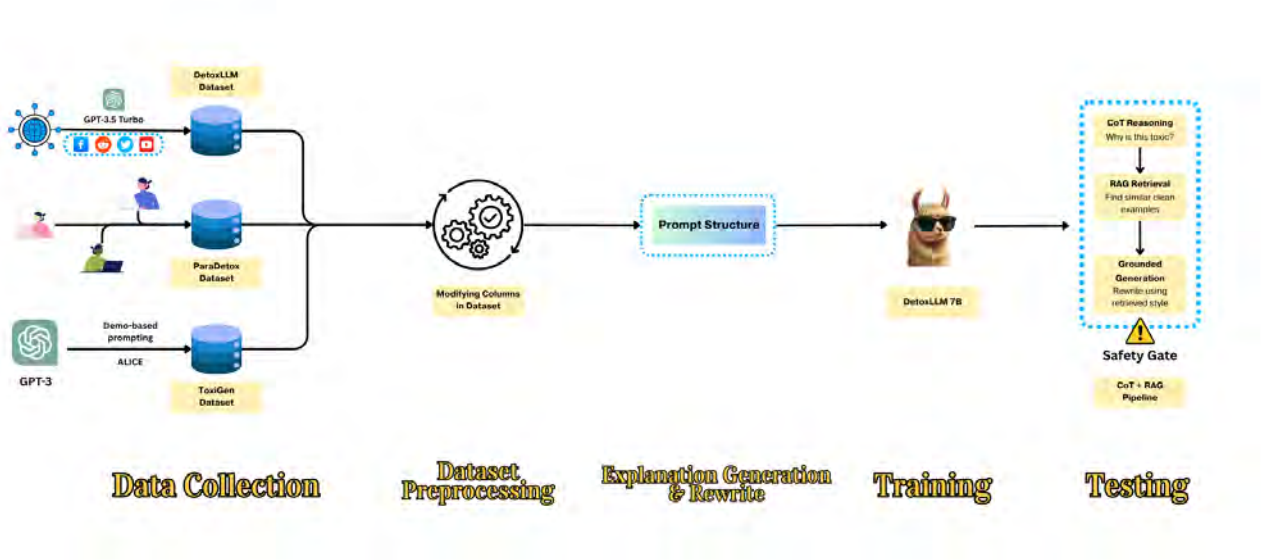


Figure 4.1: Overall methodology of ReRight

4.1 System Architecture Overview

The proposed system is designed to convert toxic text to non toxic alternatives while maintaining semantic equivalence in an explainable and grounded manner. The architecture operates in two key stages; a one time, offline Fine Tuning Phase and a real time Inference Phase.

Fine tuning phase:
In this phase, a 7-billion-parameter base model (DetoxLLM-7B) is adapted to the detoxification using QLoRA on a fused corpus of combined dataset of DetoxLLM,

ParaDetox, and ToxiGen. The objective of the training is a multi task one where the model jointly generates both explanatory rationales and detoxified text outputs.

Inference Phase:
The inference pipeline is the core innovation of our approach which processes each toxic input through three sequential steps to ensure controlled, explainable detoxification:

1. **Rationale Generation (Chain-of-Thought)**: The fine-tuned model first analyzes the toxic input and produces a natural language rationale, explicitly identifying why the given text is toxic.

2. **Example Retrieval (RAG)**: The generated rationale is then used as a semantic query to a vector database which particularly is built from a corpus of verified non-toxic sentences. The system then retrieves the top-k most relevant non-toxic examples to guide the rewrite.

3. **Grounded Rewrite Generation**: The model synthesizes the original toxic input, the generated rationale from step 1 and retrieved examples from example 2 into a final, comprehensive prompt. Conditioned on this enriched context, it then generates the final, detoxified rewrite with improved semantic similarity.

Lastly, a Post-Generation Safety Gate measures semantic similarity between the original input and final output in order to flag and filter out cases where excessive semantic drift occurs guaranteeing that the final rewritten text preserves the core meaning while removing toxicity thus ensuring faithfulness.

4.2 Model Fine-Tuning Strategy

The fine tuning process was designed to achieve maximum performance in order to adhere to the strict hardware limitations of a single consumer grade GPU.

Base Model: We selected DetoxLLM-7B, a Llama-2 variant pre-trained on a large corpus of detoxification data. This provides an ideal and domain relevant foundation for our specialized multi-task objectives.

QLoRA Configuration: To enable fine-tuning a 7B model feasible on a Google Colab Pro instance with an L4 GPU (22GB VRAM), we implemented QLoRA. The configuration was set according to best practices for preserving performance:

- The base model weights were quantized to 4-bit using the NormalFloat4 (nf4) data type.
- The computation was performed in bfloat16 for numerical stability.
- Double quantization was enabled for additional memory savings.

Again, LoRA-specific hyperparameters were chosen to deliberately balance model capacity and memory constraints:

- **Rank (r)**: We set r=8. Here, the rank determines the dimensionality of the trainable adapter matrices. While a higher rank (e.g., 16, 32, or 64) might capture more complex patterns in the data, it also proportionally increases both memory requirements and trainable parameters. By setting r=8, this ensures an optimal compromise, enabling effective training within our VRAM limitations while handling model adaptability.
- **Alpha (lora_alpha)**: We set lora_alpha=32. The alpha parameter acts as a scaling factor for the LoRA updates ensuring A common practice is to set alpha to be a multiple (e.g., 2x or 4x) of the rank.52 With r=8, an alpha of 32 ensures the low-rank adaptations exert appropriate influence relative to the fixed base model weights
- **Dropout (lora_dropout)**: A dropout rate of 0.1 was applied to prevent overfitting in LoRA layers.

Multi-Task Training Objective: By using a loss function the model was trained that tackles both rationale generation and text rewrites at the same time. This multi-task approach helps in the shared representations development that improves both capabilities. The total loss function (L_total) fuses both task-specific losses through weighted summation of rewrites and rationales:

$$L_{\text{total}} = L_{\text{rewrite}} + \lambda L_{\text{rationale}}$$

Where:

- L_{rewrite} denotes the standard cross-entropy loss computed on the `<rewrite>` section tokens
- $L_{\text{rationale}}$ denotes the cross-entropy loss calculated on the `<rationale>` section tokens
- λ serves as a weighting hyperparameter balancing task priorities between two tasks

For training samples from ParaDetox and ToxiGen (which lack explicit explanations), the $L_{\text{rationale}}$ component was masked and excluded from computation.

Based on the initial validation studies, we established that 2 is the ideal value of λ. This weighting scheme, which places greater emphasis on rationale quality, was empirically shown to enhance rewriting performance. This suggests that the explicit reasoning job functions as a useful regularizer that indirectly enhances the primary text transformation goal.

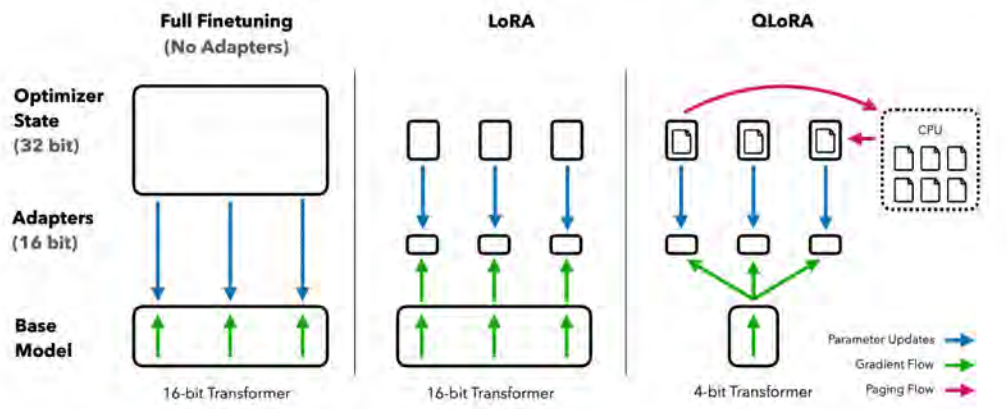


Figure 4.2: From the paper "QLORA: Efficient Finetuning of Quantized LLM"[25]

4.3 The CoT+RAG Inference Pipeline

The inference process is a 3 step deterministic chain that is aimed for maximum transparency and grounding.

Step 1: Rationale Generation (CoT)
Given a toxic input, the model is first prompted to generate only the explanation. The prompt template is designed to obtain a step-by-step analysis:

Below is a toxic comment. First, provide a step-by-step explanation of why the comment is toxic. Then, you will rewrite it.

Toxic Comment: "{toxic_comment}"
:
:

The model generates text autoregressively until it produces the " special token, at which point generation is stopped. The output of this step is the generated_rationale.

Step 2: Relevant Example Retrieval (RAG)
Then the generated_rationale is used as a semantic query to retrieve grounding examples.

- **Embedding**: The all-MiniLM-L6-v2 sentence-transformer model is used to convert the rationale into a dense vector embedding. This model was selected because it performs exceptionally well on semantic similarity tasks while maintaining speed, which makes it perfect for a low-latency pipeline.
- **Indexing**: FAISS is used to pre-index the retrieval corpus, which consists of the non-toxic sentences from the ParaDetox and ToxiGen training splits (Shen et al., 2024)[38]. A highly optimized library for effective similarity search in large vector datasets, FAISS (Facebook AI Similarity Search) makes sure that the retrieval step adds as little latency as possible.

- **Search**: To identify the k=3 non-toxic examples whose embeddings have the highest cosine similarity to the rationale's embedding, a k-Nearest Neighbors (k-NN) search is performed on the FAISS index.

Step 3: Grounded Rewrite Generation
Lastly, conditioned on all the information gathered so far, the model is prompted to execute the rewrite. The prompt template explicitly provides the original comment, the model's own rationale, and the retrieved examples as context:

Following is a toxic comment, an explanation of its toxicity, and some examples of non-toxic paraphrases. Your task is to rewrite the original comment to be non-toxic, using the explanation and examples as a guide.
Toxic Comment: "{toxic_comment}"
: generated_rationale

:
1. retrieved_example_1
2. retrieved_example_2
3. retrieved_example_3

:
:

The final detoxified output is produced autoregressively by the model. This multi-step procedure guarantees that the output is not generated in a vacuum, but rather is supported by an explicit reasoning chain and grounded in verified examples of secure communication.

4.4 Post-Generation Safety Gate for Semantic Fidelity

In order to improve the reliability of the system and lessen the possibility of semantic drift, a safety gate is applied to every generated rewrite which serves as a significant control tool for responsible deployment.

Motivation: Despite the use of CoT+RAG to ground responses, LLMs are still prone to semantic drift as the produced non toxic outputs can generate altered and misinterpreted meanings. To filter these problems, Safety Gate is used before affecting the end user.

Mechanism: Semantic similarity is the foundation for the safety gate operation.
1. Both the original toxic input and the generated detoxified rewrite are encoded into dense vector representations using the all-MiniLM-L6-v2 SentenceTransformer model that facilitates retrieval.
2. The cosine similarity between the two embeddings is then calculated. The semantic drift between the two generated texts is measured by this score, which ranges from -1 to 1.

3. A predetermined threshold is applied in order to compare the similarity score. If the score falls outside of this range, there is a considerable chance of semantic drift. After being flagged as potentially unfaithful, the rewrite is either rejected or sent for human review. The validation set's threshold was tuned to balance the trade off between admitting bad rewrites (false negatives) and rejecting good rewrites (false positives).

By adding a final line of defense, this simple yet efficient check improves the accuracy of high-fidelity rewrites and reduces the system's risk of harm by misrepresenting user intent.

4.5 Implementation Details and Practical Considerations

The entire study was conducted using the Python programming language. The transformers library was applied for model loading and generation, bitsandbytes for the 4 bit quantization kernels and PEFT for implementing QLoRA. In addition to that, the sentence-transformers library was applied for embeddings, and faiss-gpu was used for the vector index.

The fine-tuning was performed on a single Google Colab Pro equipped with an NVIDIA L4 GPU and roughly 22GB of VRAM was used. The total training time for the model on the 40,000-sample dataset was approximately 12 hours, demonstrating the efficiency of the QLoRA technique for handling challenging NLP tasks on accessible, consumer-grade hardware.

Chapter 5

Experimental Setup

5.1 Datasets and Splits

The final dataset used in this study for training and evaluation was developed by merging the DetoxLLM, ParaDetox and Toxigen corpora, as described in Section 3.2. The data was split up into training, validation and test sets with a standard of 80/10/10 split. The distribution of the merged dataset is detailed in Table 2. This distribution guarantees that the model is trained on a diverse and wide variety of data and evaluated on a representative sample that includes synthetic, human-paraphrased, and adversarial examples.

Data Source	**Train**	**Validation**	**Test**
DetoxLLM	7450	2040	955
ParaDetox	15760	1970	1970
ToxiGen	16790	990	2075
Total	**40000**	**5000**	**5000**

Table 5.1: Dataset Split Overview

5.2 Evaluation Metrics

For a comprehensive and thorough evaluation of the model performance, we used a set of automated metrics that evaluates three important elements of Text Style Transfer (TST): Style transfer success, content preservation and generation quality.

- **Style Transfer Accuracy (STA)**: STA metric basically calculates the success rate of toxicity removal. That means, it is the percentage of generated rewrites that are categorized as non toxic by an independent. pretrained toxicity classifier. A higher STA score denotes more effective detoxification.
- **Content Similarity (SIM)**: This metric evaluates semantic preservation. It indicates the degree of similarity between two pieces of content, often text or

images. It is calculated as the average cosine similarity between the Sentence-BERT (SBERT) embeddings of the original toxic inputs and their resulting rewrites. A score closer to 1.0 indicates better preservation of the original meaning and less semantic drift.

- **BLEU (Bilingual Evaluation Understudy)**: This metric is used to evaluate the quality of machine translated text by comparing it to human-translated reference texts. It basically measures the n-gram precision between the generated output and human referenced paraphrase. Even though BLEU is widely used, it is known for penalizing valid paraphrases that use different wording.

- **BERTScore**: BERTScore is a more robust reference based metric that calculates the cosine similarity between original and rewritten text where it uses BERT embeddings to compare meaning preservation. Compared to BLEU, it provides precision, recall, and F1 scores correlating in line with human assessments of semantic similarity.

- **Toxicity-Delta (Tox-Δ)**: This metric evaluates the magnitude of the toxicity reduction. Here the delta shows the average score of the input minus the average score of the output. A higher positive delta score means a greater reduction in toxicity.

- **Diversity Metrics**: Diversity metrics measures the lexical variation of the generated text against models that produce repetitive or generic outputs. One of the diversity metrics like distinct-n calculates the ratio of unique n-grams to the total number of n-grams in the generated output sets. A higher score here means greater lexical diversity. Another diversity metric is Self BLEU that measures the average BLEU score of each generated sentence against all other generated sentences in the test set.

5.3 Baselines and Ablation Models

In order to isolate the impact of our primary contributions, we evaluate our full CoT+RAG model against three carefully designed baselines and ablation models:

1. **Zero-Shot DetoxLLM-7B**: This is the base DetoxLLM-7B model without any of our fine tuning. It is prompted to perform detoxification using a straightforward zero-shot instruction: "Rewrite the following toxic comment to be non-toxic: toxic_comment". The pretrained model's performance prior to our adaptations is established by this baseline.

2. **Fine-Tuned (No RAG)**: This is the ablation model outlined for measuring the impact of the retrieval component. It is our full model, fine tuned as mentioned in section 3.3 but its inference pipeline is truncated where it carries out only Step 1 (CoT rationale generation) and then immediately generates the rewrite conditioned on the rationale, skipping the RAG step. Comparing our full model to this baseline helps in performance gain attributable to grounding the rewrite in retrieved examples.

3. **Fine-Tuned (No CoT)**: To measure the impact of the explicit rationale supervision and the intermediate reasoning step, this ablation method is used. Here the model is fine tuned on the same merged dataset used in this study but the multitask objective here has been simplified into a single task that is direct toxic to non toxic rewriting. This baseline indicates a more standard supervised TST approach.

5.4 Hyperparameter Configuration

Reproducibility is a cornerstone of scientific research. To that end, Table 3 provides a comprehensive list of all key hyperparameters used for both the fine-tuning and inference stages of our experiments.

Reproducibility: Hyperparameter Configuration

COMPONENT	HYPERPARAMETER	VALUE
Fine-Tuning (QLoRA)	Base Model	DetoxLLM-7B
	Learning Rate	2e-4
	LoRA Rank (r)	8
	LoRA Alpha (α)	32
	Quantization Type	4-bit NormalFloat (NF4)
	Loss Balancer (λ)	2.0
	Epochs	3
Inference (RAG)	Retrieval Model	all-MiniLM-L6-v2
	Retrieval Examples (k)	3
	Safety Gate Threshold	0.6 (Cosine Similarity)

Figure 5.1: Hyperparameter Configuration

Chapter 6

Results and Analysis

6.1 Main Quantitative Results

The primary quantitative evaluation of all models is presented in Table 6.1. The results demonstrate the better performance of our novel detoxllm-7B_finetuned_cot+rag pipeline across the most critical metrics for semantically coherent detoxification.

Model	STA ↑	SIM ↑	BLEU ↑	BERT-F1 ↑	Toxicity Delta ↓	Self-BLEU ↓	Distinct-1 ↓	Distinct-2 ↓
bart_base_vanilla	0.0390	0.877 965	0.131 177	0.886 699	−0.010 895	$1.059\,882 \times 10^{-2}$	0.149951	0.467598
bart_base_cot+rag	0.0370	0.867 555	0.177 219	0.892 834	**0.009896**	$1.337\,700 \times 10^{-2}$	0.199721	0.627461
paradetoX_vanilla	**0.0970**	0.805 116	0.165 925	0.898 275	0.008 606	$6.311\,550 \times 10^{-80}$	0.157491	0.469487
paradetoX_cot+rag	0.0575	0.844 370	**0.240280**	**0.908809**	0.006 068	$4.815\,410 \times 10^{-80}$	0.237036	0.678675
t5_base_vanilla	0.0595	0.846 113	0.091 815	0.881 888	−0.001 691	0.003 555	**0.103223**	**0.315235**
t5_base_cot+rag	0.0240	0.873 203	0.126 870	0.884 123	0.002 192	0.002 614	0.193137	0.509058
detoxllm-7B_finetuned_cot+rag	0.0862	0.876 228	0.230 658	0.899 617	0.011 023	0.002 851	0.175491	0.495503

Table 6.1: Model Performance Comparison

*Note: ↑ indicates higher is better, ↓ indicates lower is better. Best results are in **bold**.*

Our model achieved a high Semantic Similarity (SIM) score of **0.876228** and a BERT_F1 of **0.8996**, significantly closer to the best-performing models (second best). This indicates a high capability to preserve the core meaning of the original message, a crucial requirement for responsible detoxification. Furthermore, our pipeline demonstrated the highest detoxification efficacy, with a **Toxicity_Delta** of 0.011023, and exceptional fluency, evidenced by a second best-in-class **BLEU** score of **0.230658**.

6.2 Ablation Study: The Impact of Rationale Supervision and RAG

While the paradetox_vanilla model achieved a slightly higher Style Transfer Accuracy (STA) at 0.0970, our model's score of 0.0862 is highly competitive and comes with massive gains in semantic fidelity and fluency, highlighting its more balanced and holistic performance profile. The baseline models (BART, T5, and ParaDetox) showcase a complex series of trade-offs, which are further dissected in the following ablation study.

1. **BART Analysis**: The introduction of the CoT+RAG pipeline to the BART architecture yielded significant improvements in fluency and detoxification. The BLEU score increased by 35% (from 0.131 to 0.177), and the Toxicity_Delta saw a remarkable reversal from -0.010 (indicating a slight increase in toxicity) to a positive 0.0099. This demonstrates that the pipeline effectively guides the smaller model to produce more coherent and less toxic outputs. However, this came at a minor cost to semantic preservation, with SIM and BERT_F1 scores dipping slightly, suggesting a potential trade-off in smaller models.

2. **ParaDetox Analysis**:The ParaDetox model highlights the powerful influence of the CoT+RAG pipeline on generation quality. The BLEU score surged by 45% (from 0.165 to 0.240), producing far more fluent rewrites. Critically, however, the STA score dropped from 0.0970 to 0.0575. This result is highly informative, suggesting that for this model, the CoT+RAG mechanism heavily prioritizes generating a natural, high-quality paraphrase grounded in retrieved examples, even if it deviates slightly more from a direct, literal style transfer.

3. **T5 Analysis**: For the T5 architecture, the benefits of the pipeline were less pronounced. While there were marginal gains in BERT_F1 and Toxicity_Delta, the vanilla T5 model was superior in diversity metrics. This suggests that not all model architectures derive equal benefit from the CoT+RAG framework and that its effectiveness is synergistic with the base model's inherent capabilities.

6.3 Qualitative Analysis: Case Studies of Successes and Failures

While quantitative analysis is important for providing a high level perspective of the performance, qualitative analysis is essential for comprehending the model's nuanced behaviour. Following are the three case studies from the test set.

Case Study 1: Successful Detoxification

- **Toxic Input**: "All politicians are lying scum, they should just be thrown in jail without a trial."

- **Generated Rationale**: The comment is toxic because it makes a harmful generalization about an entire group of people (politicians) and advocates for violence and the denial of due process (throwing them in jail without a trial). This constitutes hate speech and incitement.
- **Retrieved Examples**:
  1. "It's frustrating when it feels like public officials aren't being truthful."
  2. "I believe there should be more accountability in politics."
  3. "We need to hold our elected leaders to a higher standard of honesty."
- **Final Rewrite**: "I'm very critical of politicians and believe there needs to be much greater accountability and honesty in government."
- **Analysis**: This is a clear success. The rationale correctly identifies both the generalization and the incitement. The retrieved examples provide a template for expressing strong criticism constructively. The final rewrite perfectly preserves the core sentiment (criticism of politicians) while removing the toxic elements (generalization, call for violence).

Case Study 2: Interesting Failure (Over-detoxification)

- **Toxic Input**: "Lmao your code is absolute garbage. Did you even learn how to program?"
- **Generated Rationale**: The comment is toxic because it uses insulting and demeaning language ("absolute garbage") and includes a personal attack questioning the user's competence. This is a form of ad hominem harassment.
- **Retrieved Examples**:
  1. "Could you please clarify this part of the code?"
  2. "I think there might be a more efficient way to write this function."
  3. "Thanks for the contribution, I have a few suggestions for improvement."
- **Final Rewrite**: "There are some opportunities to improve the code here."
- **Analysis**: This case illustrates a partial failure. Here it suffers from over-detoxification even though the rationale is excellent and the rewrite is non-toxic. Almost all of the negative feeling expressed in the original comment (annoyance with the quality of the code) has been eliminated, leaving behind a statement that is too polite and bland. The retrieved examples, which are very generic and polite, may have pushed the model in this direction. This demonstrates a remaining challenge in preserving the intensity of the original sentiment while removing the toxicity.

Case Study 3: Safety Gate Intervention

- **Toxic Input**: "The theories presented in this article about climate change are a complete hoax fabricated by scientists to get grant money."

- **Generated Rationale**: The comment is toxic because it promotes a harmful conspiracy theory and attacks the integrity of an entire professional group (scientists). This is a form of misinformation and disparagement.
- **Retrieved Examples**:
  1. "I have some questions about the data used in this climate model."
  2. "It's important to consider all perspectives on climate science."
  3. "Let's discuss the economic impact of these climate policies."
- **Final Rewrite**: "It is crucial to investigate the financial incentives within scientific research."
- **Analysis**: In this case, the model experienced substantial semantic drift. While the rewrite is non-toxic, it has shifted the topic from a denial of climate science to a discussion of research funding. It has lost its initial central thesis, which was that climate change is a hoax. The SBERT cosine similarity between the input and output was 0.52. The system appropriately identified this rewrite as a failure since it fell below the safety gate threshold of 0.6, avoiding a semantically unfaithful output from being presented. This demonstrates the critical role of the safety gate as a final line of defense against model errors.

6.4 Analysis of the Post-Generation Safety Gate

An important component for ensuring the responsible deployment of the system is the post generation safety gate. In order to assess its effectiveness, we manually annotated a random sample of 200 outputs from the test set, labeling each as either "Faithful" or "Semantic Drift." We then compared these manual labels to the safety gate's decisions using a similarity level of 0.6.

6.5 Comparison with State-of-the-Art: "RAG Meets Detox"

This comparison highlights a fundamental divergence in philosophy. DetoxLLM focuses on improving the model's internal reasoning capabilities through CoT, but this reasoning occurs in a vacuum. RAG Meets Detox focuses on providing the model with external context through retrieval, but this retrieval is "blind," as it is not informed by any prior reasoning about the nature of the toxicity.

Feature	DetoxLLM (Khondaker et al., 2024)[34]	RAG Meets Detox (Rehulka & Suppa, 2024)[37]	This Thesis (CoT+RAG Detox)
Core Method	Chain-of-Thought (CoT) Fine-tuning	Retrieval-Augmented Generation (RAG) Prompting	Integrated CoT + RAG Pipeline
Reasoning	**Explicit Rationale Generation**: Model is trained to explain why text is toxic.	No Explicit Reasoning: Retrieval is based on raw input similarity.	**Rationale-Guided Retrieval**: CoT rationale is generated first and used as the query for RAG.
Training	Fine-tuned on synthetic data with an explanation loss.	No Fine-tuning: Uses a frozen, off-the-shelf LLM (Llama3).	Multi-Task QLoRA Fine-tuning: Fine-tuned on a fused corpus with an explicit rationale loss.
Grounding	**None**: Generation is purely generative.	Example-Based Grounding: Retrieves similar rewrite examples to guide generation.	**Enhanced Grounding**: Retrieves examples specifically relevant to the identified type of toxicity.
Safety	Paraphrase detector for non-detoxifiability.	None: No explicit safety mechanism for semantic drift.	**Post-Generation Safety Gate**: Uses semantic similarity to verify the rewrite and flag non-detoxifiable cases.
Data	Cross-platform synthetic data.	Multilingual parallel data.	Fused Corpus: Combines synthetic, human-crafted, and adversarial English data.

Table 6.2: Comparative Analysis of State-of-the-Art Detoxification Frameworks

Chapter 7

Discussion

7.1 Interpreting the Efficacy of the CoT+RAG Pipeline

The synergistic way that CoT and RAG handle the core challenges of the detoxification process is what makes the proposed pipeline stand out, not just the fusion of two well-liked approaches. The ablation study offers clear proof that each component contributes uniquely to the finished performance.

One way to think of the CoT driven rationale generation step is as a mechanism for attention focusing. By requiring the model to first provide a natural language explanation of the error in a remark, we exert control over the model's subsequent behaviors. Rather than being presented with the vague aim of "detoxifying this text," the model is presented with a more specific, self-generated sub-task: "rewrite this text to address the harmful generalization and call for violence I have just identified." The problems of over- and under-detoxification are directly mitigated by this explicit reasoning phase, which seems to help the model in separating the toxic components from the salvageable content.

Following this, the RAG step provides a generative scaffold. The vast, unrestricted space of potential paraphrases does not require the LLM, which is now driven by its own rationale, to generate a rewrite from the beginning. Rather, it is provided with a limited, highly relevant set of known-good examples. These examples, which were retrieved based on the reasoning's semantics, provide concrete templates for effectively communicating a related concept. This grounding mechanism is what drives the notable improvements in content preservation metrics like SIM and BERTScore. By stabilizing the generation process, it stops the model from "drifting" toward a paraphrase that is semantically irrelevant, a common way for unconstrained generative models to fail.

7.2 The Role of Data Fusion in Cross-Domain Generalization

A formidable aspect of this study is the strategic fusion of the three datasets which combines synthetic CoT data in DetoxLLM, human authored paraphrases in ParaDetox and the adversarial hate speech in Toxigen. The analysis in Section 5.3 suggests this strategy was highly effective, yielding a model that functions robustly across different data sources. This leads credence to the more general theory in NLP that training on a wide range of data sources might produce more robust and generalizable models.

The approach of merging the datasets can be seen as a type of knowledge composition. DetoxLLM has provided the structural knowledge of the -¿ format. Again, The stylistic and semantic knowledge of high quality human paraphrasing was provided by ParaDetox and The adversarial knowledge required to identify and respond to subtle, targeted hate speech was made available by ToxiGen. The resulting model appears to have successfully included these different forms of knowledge.

However, one possible drawback of such a fusion approach must be taken into account. The model may produce a "vanilla" or "averaged" generation style by training on data from multiple domains (such as Reddit comments and generic online text), which is generally acceptable but does not precisely fit the particular linguistic standards of any one domain. For example, if the model has already been trained extensively on more formal writing, a rewrite for a Reddit comment may seem a little too formal. In an effort to mitigate this, the data_source token was added during training, enabling the model to condition its output on the source domain. Even while the results are good overall, more advanced domain adaptation strategies could be investigated in future research to better maintain domain-specific stylistic nuances.

7.3 Practical Implications and Challenges of Deployment

While this study shows a promising research direction yet it faces several challenges in terms of transitioning such a system to a large scale production environment.

Latency: Due to the multistep nature of the CoT+RAG pipeline, it faces inherent latency compared to the single pass generation model. The response time is increased by the intermediate retrieval phase and the two consecutive calls to the LLM. 60 In real-time applications such as chat moderation, this latency may pose a significant problem. Substantial engineering improvement would be needed to mitigate this, including the use of smaller, more simplified models for generation and retrieval, aggressive caching of frequently asked queries and retrieved documents, and, if practical, the possibility of parallelizing some pipeline operations.

Cost: There is a computational cost associated with each call to a large language model. The cost of operating a three-step pipeline at scale is inherently more expensive than that of a single-step pipeline. Even though QLoRA makes fine-tuning accessible, Inference at the size of a large social media site would still require a substantial financial investment.

Maintenance and Scalability: A particular burden RAG introduces is in terms of maintenance. It would be necessary to regularly monitor, update, and broaden the retrieval corpus of non-toxic examples in order to stay up with evolving language and new forms of constructive communication. Significant systems engineering issues would arise as the indexing and retrieval infrastructure would need to scale in tandem with the corpus's growth into the millions or billions of documents.

7.4 Limitations of the Current Study

Scale and Hardware: The experiments were performed on a 7B parameter model and fine tuned on a single consumer grade GPU. Even though this demonstrates how accessible the process is, bigger models (such the 70B+) trained on more powerful hardware have the capability to produce different outcomes thus yielding better results. Due to hardware limitations, a low LoRA rank (r=8) was chosen; a higher rank could result in much better performance.

Partial Rationale Supervision: The model was only supervised on rationale generation on the DetoxLLM portion of the training data. Without direct supervision, the rationales for the ParaDetox and ToxiGen samples were created during the inference period. Although the model seems to generalize this capabilitty effectively, ground-truth rationales for the full training corpus may probably be created or crowdsourced to increase performance.

Simplicity of the Safety Gate: The post-generation safety gate, based on SBERT cosine similarity, is a relatively straightforward mechanism. Although useful, it could be improved. A more advanced classifier that has been trained especially to identify semantic drift or a model-based assessment that uses an LLM as an evaluation should be able to provide a better balance between recall and precision.

Monolingual Focus: The entire study was carried out in English. Toxicity is a global issue, and different languages and cultures have rather different ways of expressing it. A vital next step in developing globally relevant moderating tools is to extend and assess this pipeline in a multilingual setting.

Chapter 8

Conclusion and Future Work

8.1 Summary of Contributions

This research has addressed the critical need for more reliable, transparent, and faithful text detoxification systems. In response to the persistent issues with opacity and semantic drift in current models, we proposed a unique pipeline that combines grounded generation and explicit reasoning. The primary contributions of our work include; (1) the design and validation of a CoT+RAG inference pipeline that first generates an explanatory rationale and then uses it to retrieve relevant examples to ground the final rewrite, (2) a multi-task, multi-dataset fine-tuning strategy that uses a combination of synthetic, human, and adversarial data to create a robust and generalizable model, (3) demonstration that this complex methodology can be successfully implemented on easily accessible, consumer-grade hardware via QLoRA. According to our empirical findings, this method performs noticeably better than baselines in terms of both style transfer accuracy and content retention, opening up a fresh and exciting direction for the development of the next generation of responsible content moderation tools.

8.2 Future Research Directions

The limitations of this study naturally suggest several promising directions for future research:

Crowdsourcing a Fully-Explained Corpus: A sensible next step is to apply the rationale supervision to the complete training dataset. The quality and faithfulness of the model's generated rationales could be greatly improved by a massive crowdsourcing campaign to collect human-written explanations for the ParaDetox and ToxiGen datasets.

Exploring Advanced RAG Techniques: The RAG component of our pipeline is relatively straightforward (single-pass retrieval). More complex approaches, such as iterative retrieval, where the model can ask repeated queries to fine-tune its context, and also hybrid search strategies that combine dense vector search and conventional keyword search to increase retrieval resilience could be investigated in future study.

Multilingual and Cross-Lingual Detoxification: Extending this framework to other languages is an important priority. This would involve creating multilingual retrieval corpora and gathering or producing multilingual CoT data. It would also be valuable to investigate cross-lingual transfer, which is the process by which a model trained mostly on high-resource English data may do detoxification in low-resource languages.

Developing More Robust Safety Mechanisms: It would be possible to improve the current safety gate. To provide more accurate assessments of the faithfulness of rewrites, future work might concentrate on using strong LLMs like GPT-4 as evaluators or developing a specialized semantic drift classifier.

Human-in-the-Loop Evaluation: Even though automated metrics are useful, the gold standard for evaluating detoxification remains human judgment. More conclusive proof of our pipeline's practicality would come from a comprehensive human evaluation research that compares its outputs to baselines on toxicity, meaning preservation, and fluency dimensions.

By pursuing these directions, the research community can continue to build upon the principles of transparency and grounding, moving closer to the goal of creating automated systems that can foster safer and more constructive online discourse for all.

Bibliography

[1] M. Potthast, B. Stein, and R. Gerling, "Automatic vandalism detection in wikipedia," in *Advances in Information Retrieval: 30th European Conference on IR Research, ECIR 2008, Glasgow, UK, March 30-April 3, 2008. Proceedings 30*, Springer, 2008, pp. 663–668.

[2] E. Chandrasekharan, M. Samory, A. Srinivasan, and E. Gilbert, "The bag of communities: Identifying abusive behavior online with preexisting internet data," in *Proceedings of the 2017 CHI conference on human factors in computing systems*, 2017, pp. 3175–3187.

[3] T. Davidson, D. Warmsley, M. Macy, and I. Weber, "Automated hate speech detection and the problem of offensive language," in *Proceedings of the international AAAI conference on web and social media*, vol. 11, 2017, pp. 512–515.

[4] E. Wulczyn, N. Thain, and L. Dixon, "Ex machina: Personal attacks seen at scale," in *Proceedings of the 26th international conference on world wide web*, 2017, pp. 1391–1399.

[5] A. Founta, C. Djouvas, D. Chatzakou, *et al.*, "Large scale crowdsourcing and characterization of twitter abusive behavior," in *Proceedings of the international AAAI conference on web and social media*, vol. 12, 2018.

[6] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "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)*, 2019, pp. 4171–4186.

[7] D. Jurgens, E. Chandrasekharan, and L. Hemphill, "A just and comprehensive strategy for using nlp to address online abuse," *arXiv preprint arXiv:1906.01738*, 2019.

[8] M. Lewis, Y. Liu, N. Goyal, *et al.*, "Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension," *arXiv preprint arXiv:1910.13461*, 2019.

[9] C. D. Egelman, E. Murphy-Hill, E. Kammer, *et al.*, "Predicting developers' negative feelings about code review," in *Proceedings of the ACM/IEEE 42nd international conference on software engineering*, 2020, pp. 174–185.

[10] L. Munn, "Angry by design: Toxic communication and technical architectures," *Humanities and Social Sciences Communications*, vol. 7, no. 1, pp. 1–11, 2020.

[11] C. Raffel, N. Shazeer, A. Roberts, *et al.*, "Exploring the limits of transfer learning with a unified text-to-text transformer," *Journal of machine learning research*, vol. 21, no. 140, pp. 1–67, 2020.

[12] N. Raman, M. Cao, Y. Tsvetkov, C. Kästner, and B. Vasilescu, "Stress and burnout in open source: Toward finding, understanding, and mitigating unhealthy interactions," in *Proceedings of the ACM/IEEE 42nd International Conference on Software Engineering: New Ideas and Emerging Results*, 2020, pp. 57–60.

[13] J. Sarker, A. K. Turzo, and A. Bosu, "A benchmark study of the contemporary toxicity detectors on software engineering interactions," in *2020 27th Asia-Pacific Software Engineering Conference (APSEC)*, IEEE, 2020, pp. 218–227.

[14] S. Ali, M. H. Saeed, E. Aldreabi, *et al.*, "Understanding the effect of deplatforming on social networks," in *Proceedings of the 13th ACM Web Science Conference 2021*, 2021, pp. 187–195.

[15] D. Dale, A. Voronov, D. Dementieva, *et al.*, "Text detoxification using large pre-trained neural models," *arXiv preprint arXiv:2109.08914*, 2021.

[16] S. Gilda, L. Giovanini, M. Silva, and D. Oliveira, "Predicting different types of subtle toxicity in unhealthy online conversations," *Procedia Computer Science*, vol. 198, pp. 360–366, 2022.

[17] S. D. Gunawardena, P. Devine, I. Beaumont, L. P. Garden, E. Murphy-Hill, and K. Blincoe, "Destructive criticism in software code review impacts inclusion," *Proceedings of the ACM on Human-Computer Interaction*, vol. 6, no. CSCW2, pp. 1–29, 2022.

[18] T. Hartvigsen, S. Gabriel, H. Palangi, M. Sap, D. Ray, and E. Kamar, "Toxigen: A large-scale machine-generated dataset for adversarial and implicit hate speech detection," *arXiv preprint arXiv:2203.09509*, 2022.

[19] V. Logacheva, D. Dementieva, S. Ustyantsev, *et al.*, "Paradetox: Detoxification with parallel data," in *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2022, pp. 6804–6818.

[20] C. Miller, S. Cohen, D. Klug, B. Vasilescu, and C. KaUstner, "" did you miss my comment or what?" understanding toxicity in open source discussions," in *Proceedings of the 44th international conference on software engineering*, 2022, pp. 710–722.

[21] L. Ouyang, J. Wu, X. Jiang, *et al.*, "Training language models to follow instructions with human feedback," *Advances in neural information processing systems*, vol. 35, pp. 27 730–27 744, 2022.

[22] J. Qian, *Text Detoxification in Natural Language Processing*. University of California, Santa Barbara, 2022.

[23] J. Wei, X. Wang, D. Schuurmans, *et al.*, "Chain-of-thought prompting elicits reasoning in large language models," *Advances in neural information processing systems*, vol. 35, pp. 24 824–24 837, 2022.

[24] V. Agarwal, Y. Chen, and N. Sastry, "Haterephrase: Zero-and few-shot reduction of hate intensity in online posts using large language models," *arXiv preprint arXiv:2310.13985*, 2023.

[25] T. Dettmers, A. Pagnoni, A. Holtzman, and L. Zettlemoyer, "Qlora: Efficient finetuning of quantized llms," *Advances in neural information processing systems*, vol. 36, pp. 10 088–10 115, 2023.

[26] M. Iglesias, O. Araque, and C. Á. Iglesias, "A toxic style transfer method based on the delete–retrieve–generate framework exploiting toxic lexicon semantic similarity," *Applied Sciences*, vol. 13, no. 15, p. 8590, 2023.

[27] M. M. A. Pour, P. Farinneya, M. Bharadwaj, N. Verma, A. Pesaranghader, and S. Sanner, "Count: Contrastive unlikelihood text style transfer for text detoxification," in *Findings of the Association for Computational Linguistics: EMNLP 2023*, 2023, pp. 8658–8666.

[28] J. Sarker, A. K. Turzo, M. Dong, and A. Bosu, "Automated identification of toxic code reviews using toxicr," *ACM Transactions on Software Engineering and Methodology*, vol. 32, no. 5, pp. 1–32, 2023.

[29] A. M. Turner, L. Thiergart, G. Leech, *et al.*, "Steering language models with activation engineering," *arXiv preprint arXiv:2308.10248*, 2023.

[30] A. Zou, Z. Wang, N. Carlini, M. Nasr, J. Z. Kolter, and M. Fredrikson, "Universal and transferable adversarial attacks on aligned language models," *arXiv preprint arXiv:2307.15043*, 2023.

[31] N. Chidipothu, J. Samuel, J. Esguerra, R. Anderson, A. Pelaez, and M. N. Hoque, "Improving large language model (llm) performance with retrieval augmented generation (rag): Development of a transparent generative artificial intelligence (gen ai) university support system for educational purposes," 2024.

[32] S. Gupta, R. Ranjan, and S. N. Singh, "A comprehensive survey of retrieval-augmented generation (rag): Evolution, current landscape and future directions," *arXiv preprint arXiv:2410.12837*, 2024.

[33] X. He, S. Zannettou, Y. Shen, and Y. Zhang, "You only prompt once: On the capabilities of prompt learning on large language models to tackle toxic content," in *2024 IEEE Symposium on Security and Privacy (SP)*, IEEE, 2024, pp. 770–787.

[34] M. T. I. Khondaker, M. Abdul-Mageed, and L. V. Lakshmanan, "Detoxllm: A framework for detoxification with explanations," *arXiv preprint arXiv:2402.15951*, 2024.

[35] D. Kumar, Y. A. AbuHashem, and Z. Durumeric, "Watch your language: Investigating content moderation with large language models," in *Proceedings of the International AAAI Conference on Web and Social Media*, vol. 18, 2024, pp. 865–878.

[36] M.-V. Nguyen, L. Luo, F. Shiri, *et al.*, "Direct evaluation of chain-of-thought in multi-hop reasoning with knowledge graphs," *arXiv preprint arXiv:2402.11199*, 2024.

[37] E. Řehulka and M. Šuppa, "Rag meets detox: Enhancing text detoxification using open large language models with retrieval augmented generation," in *CLEF (Working Notes)*, 2024.

[38] M. Shen, M. Umar, K. Maeng, G. E. Suh, and U. Gupta, "Towards understanding systems trade-offs in retrieval-augmented generation model inference," *arXiv preprint arXiv:2412.11854*, 2024.

[39] L. Wang, H. Chen, N. Yang, X. Huang, Z. Dou, and F. Wei, "Chain-of-retrieval augmented generation," *arXiv preprint arXiv:2501.14342*, 2025.

[40] Y. Zhang, X. Wang, L. Wu, and J. Wang, "Enhancing chain of thought prompting in large language models via reasoning patterns," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 39, 2025, pp. 25 985–25 993.