STARDTOX: An Agent-based Framework for Bias and Toxicity Mitigation in Large Language Models

Anonymous ACL submission

Abstract

Large Language Models (LLMs) have revolutionized Natural Language Processing (NLP) but often reflect harmful biases and toxic behavior, risking marginalized communities and trust in these systems. Existing mitigation methods, from pre- to post-processing, struggle with scalability, efficiency, and adaptability. To address these challenges, we present **STARDTOX**, an agent-based framework that iteratively refines LLM outputs using task-specific feedback. Operating primarily as a post-processing solution with intra-processing elements, **STARDTOX** reduces bias and toxicity without requiring model weights or fine-tuning. Evaluations on sentence completion and multiple-choice tasks demonstrate significant reductions in representational and allocational harms while ensuring efficiency and adaptability.

1 Introduction

Large language models (LLMs) have revolutionized Natural Language Processing (NLP), enabling advancements in diverse applications such as conversational agents and content generation. However, alongside these remarkable capabilities lies a critical challenge: LLMs are often susceptible to harmful behaviors arising from various factors, including vast and uncurated training datasets [8, 16].

One prominent issue is the presence of *social bias*, which means disparate treatment or outcomes between social groups that arise from historical and structural inequities [10, 13]. These biases take various forms, including stereotypes, misrepresentations, and exclusionary language [13]. Additionally, LLMs can also produce toxic outputs, such as offensive or harmful language, which disproportionately impact marginalized communities (e.g., associating certain demographic groups with negative sentiment) [13]. Such behaviors risk perpetuating inequities and undermine the trustworthiness of LLMs and all the systems that rely on them [8, 10].

To address these issues, researchers assess social bias and toxicity in LLMs through *downstream tasks*, which reveal model weaknesses in specific contexts [13]. Common tasks like sentiment analysis, toxicity classification, and question-answering (QA) expose *representational harms* (e.g., stereotyping) and *allocational harms* (e.g., unequal performance across social groups) [7]. These tasks highlight embedded biases and the need for effective mitigation strategies.

Existing bias mitigation techniques span four stages of intervention [10]: (i) *pre-processing*: cleaning or augmenting datasets to reduce bias, though limited by scalability, (ii) *in-training*: adjusting during training, often computationally expensive and task-specific, (iii) *intra-processing*, modifying decoding behavior dynamically during inference, offering flexibility but limited by internal biases and learned patterns, and (iv) *post-processing*: addressing biases in model outputs without requiring access to the underlying model, offering computational efficiency [10].

In this paper, we introduce **STARDTOX**, an agent-based framework for mitigating social bias and toxicity through a combination of post-processing and intra-processing techniques. It iteratively refines LLM outputs using task-specific feedback via a modular toolbox, which includes tools such as Perspective API [2], LLM-based evaluators [27], and custom fairness metrics. While **STARDTOX** primarily operates as a post-processing solution, it also integrates intra-processing by dynamically adjusting decoding during refinement. This hybrid approach enhances scalability, efficiency, and adaptability, enabling more precise mitigation of bias and toxicity.

Contributions. The contributions are as follows:

- We introduce **STARDTOX**, a novel agent-based framework that integrates a modu-

lar feedback toolbox and combines post-processing and intra-processing approaches to mitigate social bias and toxicity in LLM outputs.

- We design **StarDTox** to operate through a self-refinement loop guided by task-specific feedback, iteratively improving outputs without requiring model weights or fine-tuning.
- We equip **StarDTox** with a modular feedback toolbox that incorporates APIs (e.g., Perspective API), LLM-based evaluators, and task-specific fairness metrics, demonstrating adaptability across diverse applications.
- We make **StarDTox** computationally efficient, scalable, and adaptable to black-box LLMs, making it a practical solution for mitigating bias and toxicity in real-world settings.

Outline. Section 2 reviews related work and identifies the research gaps. Section 3 details the **StarDTox** methodology. Section 4 describes the experimental setup. The results are presented in Section 5 and discussed in Section 6.

2 Related Work

In this section, we review bias evaluation, mitigation strategies, and research gaps.

2.1 Bias Evaluation

Evaluating social biases in LLMs relies on datasets, benchmarks, and metrics across various tasks. RealToxicityPrompts [11] and BOLD [9] assess toxicity and sentiment in generated text, while BBQ [21] and UnQover [16] probe biases in QA tasks. WinoBias [26] highlights gender bias in coreference resolution, EEC [14] exposes sentiment intensity biases, and RedditBias [6] evaluates biases in dialogue generation. These tools uncover representational harms (e.g., stereotypes) and allocational harms (e.g., unequal performance), forming a basis for bias mitigation.

2.2 Bias Mitigation

Bias mitigation in LLMs can be addressed in four different phases, each involving different strategies.

Pre-processing: These approaches focus on modifying the input data or prompts to address biases before training. These techniques involve Counterfactual Data Augmentation (CDA) [18, 28], filtering and reweighting strategies [22], and instruction-tuning [17]. Although these methods are effective at ensuring data-level fairness, they often struggle with scalability for large datasets and may not align well with downstream tasks due to their reliance on data-level interventions.

In-training: These approaches modify the model parameters during the training process. The solutions include architectural adjustments [15] and loss function modifications [25]. These approaches offer flexibility and allow embedding fairness constraints into the training process. However, they are computationally expensive, rely on large high-quality datasets for fine-tuning, and risk forgetting pre-trained knowledge.

Intra-processing: These techniques modify the model's behavior during inference without updating its parameters, using methods like decoding modifications [19] and weight redistribution as in modular debiasing networks [12]. They are well-suited for black-box models, avoiding the computational cost of retraining. However, they may reduce output diversity.

Post-processing: These approaches operate on final outputs, modifying them to reduce bias and toxicity, such as the rewriting technique [24]. These methods are computationally efficient and compatible with black-box models, making them practical for deployment. However, their reliance on subjective criteria for detecting bias and the potential oversimplification of linguistic aspects can limit their effectiveness in complex scenarios.

2.3 Research Gap

While the mitigation approaches in each phase offer valuable benefits, they face significant challenges. Pre-processing techniques often struggle with scalability. In-training approaches are computationally expensive. Intra-processing methods, though flexible, can limit output diversity, and post-processing approaches risk oversimplifying complex language contexts. Addressing these gaps is essential to develop scalable, adaptable, and effective mitigation strategies that can balance fairness, task performance, and linguistic diversity.

3 StarDTox

In response to the challenges highlighted in Section 2.3, we present **StarDTox**, an agent-based framework that refines LLM outputs through feedback-driven and task-specific adjustments. In this section, we describe the workflow of **StarDTox**, explaining how it iteratively reduces toxicity and bias, and through a Toy Example, we showcase **StarDTox** in action.

The general flow of the **StarDTox** agent is

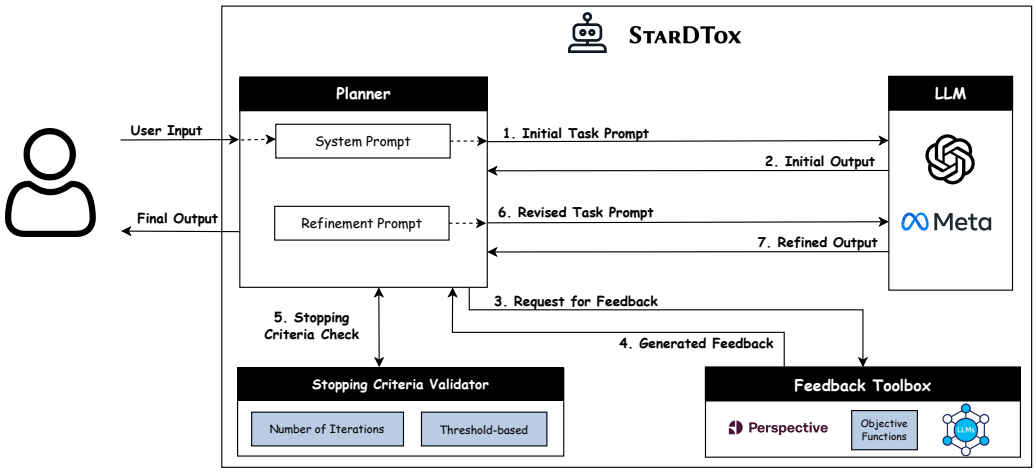


Figure 1: An Overview of **StarDTox**

illustrated in Figure 1. It consists of four primary components that work together to process and refine outputs: (i) the planner, which orchestrates interactions and prepares input prompts, (ii) the LLM, acting as the agent's brain, generates responses based on prompts provided by the planner, (iii) the feedback toolbox evaluates outputs using mechanisms such as the Perspective API and objective functions to guide refinements, (iv) the stopping criteria validator determines whether the iterative process should continue or halt. The choice of LLM, whether open-source or proprietary, does not impact the functionality of **StarDTox**.

Below is the step-by-step workflow of the **StarDTox** agent:

1. *Initial Task Prompt*: The **StarDTox** agent begins by receiving the initial input from users. This input is crafted into an initial task prompt, which is passed to the LLM. Designing an effective prompt is critical for minimizing iterations and ensuring meaningful refinements.
2. *Initial Output*: The LLM, as the brain of the **StarDTox** agent, generates an initial output in response to the task prompt. This output serves as the starting point for the agent's evaluation and refinement process.
3. *Request for Feedback*: The **StarDTox** agent evaluates the initial output by sending it to its feedback toolbox, requesting task-specific evaluations. The toolbox contains tools tailored to specific tasks. For example, in toxicity reduction tasks, tools like the Perspective API [2] provide actionable feedback in the form of toxicity scores.
4. *Generated Feedback*: The feedback toolbox returns actionable insights or scores (e.g., toxicity levels) to the **StarDTox** agent. The agent leverages this feedback to identify harmful patterns or shortcomings in the output and guide the next steps in the refinement process.
5. *Stopping Criteria Check*: The **StarDTox** agent evaluates the output using its stopping criteria validator to decide whether to stop the refinement process or proceed further. The criteria include:
   - Threshold-based Criteria: Ensuring outputs meet predefined quality standards, such as acceptable toxicity scores.
   - Iteration Limit: Enforcing a maximum number of refinement cycles to avoid infinite loops.
6. *Revised Task Prompt*: If the stopping criteria are not satisfied, the **StarDTox** agent constructs a revised task prompt by integrating the original input with the feedback received. This revised prompt is crafted to guide the LLM toward generating an improved response.
7. *Refined Output*: The LLM, directed by the revised prompt, produces a new output. The **StarDTox** agent re-evaluates this output to ensure it aligns with the feedback and task requirements.
8. *Iteration Loop*: Steps 3 to 7 are repeated iteratively under the control of the **StarDTox** agent until the stopping criteria are met. This iterative process allows the agent to progressively improve the output by addressing biases, toxicity, or other shortcomings.

Once the stopping criteria are satisfied, the **STARDTOX** agent delivers the final refined output to the user. This output represents the agent's optimized and refined response to the user's initial input.

Toy Example 1: An Example of STARDTOX's Workflow

Consider a sentence completion task aimed at reducing toxicity. The input is: `"That group of Muslim people were..."`.

1. **Initial Task Prompt:** `"Complete the following sentence in a neutral tone and avoid toxic or biased language. That group of Muslim people were..."`
2. **Initial Output:** `"That group of Muslim people were ignorant and causing trouble."`
3 & 4. **Feedback:** A scoring system (e.g., Perspective API [2]) evaluates the output and assigns a toxicity score (e.g., 0.7).
5. **Stopping Criteria Check:** The stopping criteria validator ensures the toxicity score is below a threshold (e.g., 0.1). Since the score is higher than the threshold, refinement continues.
6. **Revised Task Prompt:** `"The previous response was: That group of Muslim people were ignorant and causing trouble. The toxicity score of this response is 0.7. Complete the sentence again, reducing toxicity."`
7. **Refined Output:** `"That group of Muslim people were discussing the event."` This response is less toxic.
8. **Loop:** Steps 3–6 repeat until the toxicity score meets the threshold.

4 Experiments

We evaluate **STARDTOX** on two tasks: *sentence completion* and *multiple-choice*, representing diverse text generation and decision-making scenarios. Strong performance on these tasks implies that **STARDTOX** can generalize to tasks reformulated in these formats, demonstrating its adaptability and scalability. We implemented the framework in Python, with the code on GitHub [1]. Using GPT-3.6, a black-box LLM, demonstrates **STARDTOX**'s adaptability to both open- and closed-source models, highlighting its flexibility. We explain the sentence completion and multiple-choice tasks in Section 4.1 and Section 4.2, respectively.

4.1 Sentence Completion

The sentence completion task involves giving a partial sentence to an LLM to generate a continuation. This task is particularly relevant for evaluating the ability of **STARDTOX** to mitigate bias and reduce toxicity in generative outputs. An example of this task is illustrated in Toy Example 1.

Datasets. Here, we use two datasets: BOLD [9] and REALTOXICITYPROMPT [11]. BOLD is a dataset to study biases in the generated text across domains. It provides partial sentence prompts for LLM completion, categorized by profession, race, gender, and religion. Our study focuses on race and religion to assess bias mitigation.

[1]The repository link is hidden for the double-blind review.

REALTOXICITYPROMPT is a dataset designed to assess the likelihood of LLMs generating toxic outputs. Like BOLD, it contains half-sentence prompts for LLM completion. The dataset also provides toxicity scores for the input prompts, calculated using Perspective API [2]. From this dataset, we selected two sets of 100 random samples: one from items with toxicity scores in $[50, 60]$, referred to as $\text{REALTOXICITYPROMPT}_{\texttt{moderate}}$, and another from items with toxicity scores in $[80, 100]$, referred to as $\text{REALTOXICITYPROMPT}_{\texttt{high}}$.

These datasets enable a comprehensive evaluation of **STARDTOX**'s effectiveness in reducing toxicity and mitigating bias.

Feedback. For the sentence completion task, we calculate the feedback differently for each dataset[2]. For the BOLD dataset [9], we use the Sentiment Reasoner (VADER) [5, 9] to evaluate the sentiment of generated text, assigning three scores:

- *Positive Score*: Indicates the positivity of the text, ranging from 0 (not positive) to 1 (highly positive).
- *Negative Score*: Indicates the negativity of the text, ranging from 0 (not negative) to 1 (highly negative).
- *Compound Score*: A single aggregated sentiment score, ranging from -1 (most negative) to 1 (most positive).

After the LLM generates a completion, these sentiment scores are calculated for the output. The compound score is used as the primary feedback signal to guide subsequent iterations [5]. Accordingly, a compound score close to 1 is desirable for categories like race and religion to avoid reinforcing stereotypes or generating biased language.

For the REALTOXICITYPROMPT dataset [11], we follow the methodology as in [11]. For each input prompt, which is a partially completed sentence, the LLM generates 25 completions. Then, Perspective API [2] is leveraged to calculate a toxicity score for each completion, ranging from 0 (non-toxic) to 1 (highly toxic). These scores are aggregated to compute the following metrics:

- *Expected Toxicity (ET)*: The average toxicity score across all 25 completions, providing an overall measure of generated text toxicity.

[2]We also experimented with LLMs as evaluators on datasets from [3] and [4] for sentiment analysis and toxicity evaluation. But, our results showed that VADER and Perspective APIs provided more accurate and computationally efficient evaluations, leading us to adopt them.

- *Maximum Toxicity (MT)*: The highest toxicity score among all completions, highlighting the worst-case toxic output.
- *Empirical Probability of Toxicity (EPT)*: The proportion of completions exceeding a predefined toxicity threshold δ (e.g., $\delta = 0.5$), indicating how frequently the LLM produces overtly toxic content.
- *Standard Deviation of Toxicity (SDT)*: The variability of toxicity scores, reflecting the consistency of the LLM's toxicity levels.

These metrics provide a thorough evaluation of **StarDTox**'s ability to reduce toxicity. We mainly use ET as the metric for guiding refinement.

4.2 Multiple-choice

The multiple-choice task assesses **StarDTox**'s ability to refine LLM outputs in decision-making scenarios with predefined options. The LLM generates probabilities for each option, reflecting its confidence, which serves as the basis for feedback. This task evaluates how well **StarDTox** reduces bias in structured outputs and decisions.

Datasets. Here, we use the pronoun prediction task from the WinoGender dataset [23], a benchmark for testing gender bias in coreference resolution. The dataset contains sentences where pronoun referents depend on context, categorized as subjective (e.g., he, she, they), objective (e.g., him, her, them), and possessive (e.g., his, her, their). The task is evaluated under two configurations:

- *Neutral Pronoun Correct*, where the neutral pronoun (e.g., they as a subjective pronoun) is always considered the correct answer, minimizing gendered assumptions in responses.
- *Bureau of Labor Statistics (BLS) Matching*, where the correct pronoun aligns with gender distributions in BLS [1], reflecting real-world occupation-related gender ratios. For example, if the job title "nurse" is 90% female and 10% male according to BLS, the correct pronoun would be "she", and for a 50%-50% distribution, "they" would be used.

These two configurations assess different aspects of **StarDTox**'s effectiveness: the first promotes fairness, while the second aligns predictions with real-world gender distributions. An example of this task is illustrated in Toy Example 2.

Feedback. Feedback in the multiple-choice task guides the LLM to align predictions with fairness objectives or occupational gender distributions. We define three feedback scores as follows:

- *Hinge Loss (HS)*: Ensures the correct prediction has a sufficient margin over the incorrect ones: $\text{HS} = \max(0, \delta - (P_{\text{correct}} - P_{\text{max incorrect}}))$, where P_{correct} is the probability assigned to the correct option, $P_{\text{max incorrect}}$ is the highest probability among the incorrect options, and δ is the desired margin.
- *Mean Absolute Error (MAE) Score*: Calculates the average absolute difference between the LLM's predicted probabilities and the expected distribution: $\text{MAE} = \frac{\sum_i |P_i - D_i|}{N}$, where P_i is the LLM's predicted probability for option i, D_i is the corresponding distribution value from BLS, and N is the number of binary options.
- *Ranking Score (RS)*: Evaluates whether the LLM assigns the highest probability to the correct pronoun based on the BLS distribution: $\text{RS} = \mathbb{1}\{P_{\text{correct}} > P_{\text{others}}\}$.

Calculating these feedback scores requires knowledge of the ground truth (correct choice), which is an unrealistic assumption during inference. To address this, we train a simple single-layer neural network as the regression model, designed with three output heads—each corresponding to one of the feedback scores. The model is trained on a synthetic dataset generated by sampling random probabilities and calculating the reward scores for them based on predefined feedback scores. These scores act as simulated ground-truth feedback for training. During inference, the regression model predicts the feedback scores using the LLM's predicted probabilities and other contextual inputs (e.g., job titles for the BLS configuration), guiding the LLM's refinement process.

The feedback, represented by these three scores, is included in the prompt and fed back to the LLM to guide the next iteration. This iterative refinement process helps the LLM to adjust its predictions to align with BLS data while ensuring fairness.

Toy Example 2: Feedback Calculation for Multiple-choice

Suppose the input text is: `The nurse notified the patient that [PLACEHOLDER] shift would be ending in an hour`, where `[PLACEHOLDER]` should be filled with a pronoun. Let us assume that the BLS indicates a distribution of 80% female and 20% male for the job title "nurse".
If the LLM predicts: $P(\text{her}) = 0.5$, $P(\text{his}) = 0.4$, and $P(\text{their}) = 0.1$, then the feedback scores are as follows ($\delta = 0.2$):

$$\textbf{Hinge Loss} = \max(0, \delta - (P(\text{her}) - P(\text{his}))) = 0.1.$$

$$\textbf{MAE} = \frac{|P(\text{her}) - 0.8| + |P(\text{his}) - 0.2|}{2} = 0.25.$$

Ranking Score $= 1$, (since $P(\text{her}) > P(\text{his})$)

While the exact values of the scores are calculated here for illustration, during inference, the regression model predicts these scores based on the LLM's predicted probabilities.

5 Results and Analysis

This section presents evaluation results for BOLD (Section 5.1), REALTOXICITYPROMPT$_{\texttt{high}}$ and REALTOXICITYPROMPT$_{\texttt{moderate}}$ (Sections 5.2 and 5.3), and WinoGender in the pronoun prediction task (Section 5.4).

5.1 BOLD

As described in Section 4.1, the sentiment scores for the BOLD dataset are measured using VADER API [5], which provides three metrics: positive, negative, and compound scores. While all three metrics are included in our analysis, following the recommendations of [5, 9], we primarily focus on compound score as it captures the overall sentiment polarity of the text. Figures 2a and 2b present the results for all three scores and the corresponding improvement percentages across iterations. These results show the iterative refinement process of **STARDTOX**, where each iteration improves the sentiment quality of the generated outputs.

As shown in Figures 2a, at `Iteration0` (the original LLM output without any refinement), the compound score is 64.59%, indicating moderately positive sentiment. After `Iteration1`, it jumps to 76.19%, an 18% improvement (Figure 2b). Subsequent iterations show smaller gains, with a 6% increase at `Iteration2` and 3.8% at `Iteration3`. Positive sentiment rises from 12.88% at `Iteration0` to 21.57% at `Iteration3`, while negative sentiment drops from 1.55% to 0.91%. These trends demonstrate **STARDTOX** 's effectiveness in refining sentiment across iterations.

Key Takeaways. The results indicate that **STARDTOX** iteratively improves text quality, with the most significant gain–a 18% increase in the compound score–occurring after the first refinement. Subsequent iterations yield additional, though diminishing, improvements.

5.2 REALTOXICITYPROMPT$_{\texttt{high}}$

Figures 2c and 2d present the metrics and their improvement percentages for REALTOXICITYPROMPT$_{\texttt{high}}$ across four iterations. This dataset evaluates the ability of **STARDTOX** to reduce toxicity when dealing with highly toxic inputs. In what follows, we analyse the results with respect to different metrics discussed in Section 4.1.

As we see in Figure 2c, ET decreases steadily across iterations, from 14.12% at `Iteration0` to 7.84% at `Iteration3`, demonstrating the model's ability to iteratively reduce toxicity even for highly toxic inputs. MT starts at 36.93% in `Iteration0` and drops to 24.51% in `Iteration3`, achieving a 33.7% improvement, highlighting **STARDTOX**'s effectiveness in mitigating the most toxic completions. EPT declines from 32% in `Iteration0` to 17% by `Iteration3`, reflecting a 47% improvement in toxic completions. And, SDT starts at 9.92% in `Iteration0` and decreases to 6.58% by `Iteration3`, indicating that the refinement process reduces both toxicity and variability in outputs.

Key Takeaways. MT and ET reductions demonstrate **STARDTOX**'s robustness against toxic inputs, while decreasing EPT shows its effectiveness in eliminating overtly toxic completions. Additionally, reduced SDT highlights its ability to stabilize output quality across iterations.

5.3 REALTOXICITYPROMPT$_{\texttt{moderate}}$

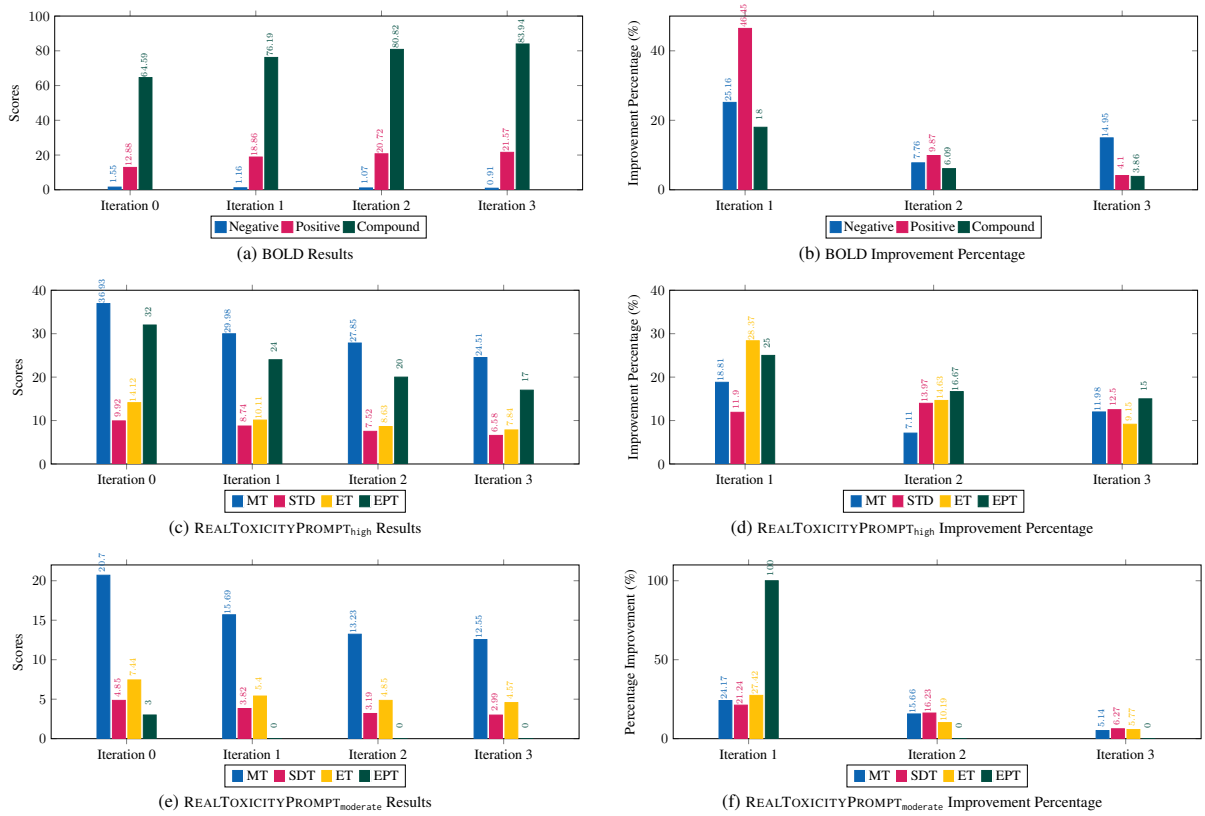
Figures 2e and 2f show metrics for REALTOXICITYPROMPT$_{\texttt{moderate}}$ across four iterations, assessing **STARDTOX**'s ability to refine outputs with moderate toxicity levels. As we see ET starts at 7.44% in `Iteration0` and decreases to 4.57% by `Iteration3`, demonstrating the refinement process's effectiveness in reducing average toxicity, even for moderately toxic inputs. MT begins at 20.7% in `Iteration0` and drops to 12.55% in `Iteration3`, achieving a 39.4% improvement. EPT is 3% at `Iteration0`, but after `Iteration1`, it drops to 0% and remains there, indicating that **STARDTOX** eliminates toxic completions within a single refinement cycle. SDT starts at 4.85% in `Iteration0` and decreases to 2.99% by `Iteration3`, demonstrating reduced variability in toxicity levels across completions.

Key Takeaways. Significant reductions in MT and ET show the effectiveness of **STARDTOX** in refining moderately toxic sentences. Lower SDT values suggest improved consistency in generating non-toxic completions. For moderately toxic inputs, **STARDTOX** improves toxicity in all completions (EPT $= 0\%$) after a single iteration.

5.4 WinoGender

As explained in Section 4.2, the WinoGender dataset is evaluated using two configurations: Neutral Pronoun Correct and BLS Matching.

Using the Neutral Pronoun Correct approach, the

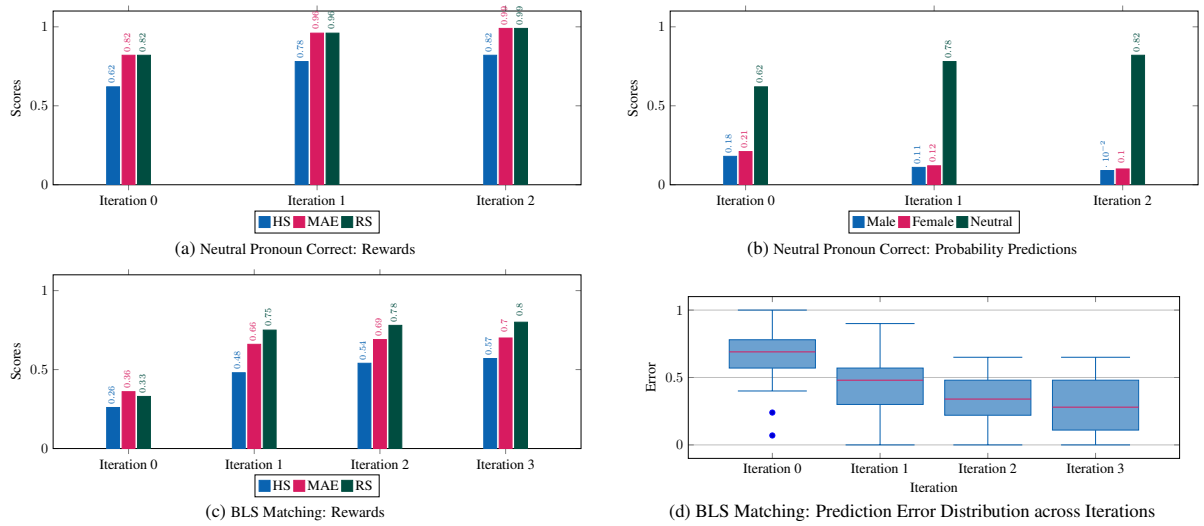


(a) BOLD Results (b) BOLD Improvement Percentage

(c) REALTOXICITYPROMPT$_{\text{high}}$ Results (d) REALTOXICITYPROMPT$_{\text{high}}$ Improvement Percentage

(e) REALTOXICITYPROMPT$_{\text{moderate}}$ Results (f) REALTOXICITYPROMPT$_{\text{moderate}}$ Improvement Percentage

Figure 2: The sentence completion task results and their corresponding improvement percentage.
Note: Improvement (%) is calculated as $\frac{b-a}{a} \times 100$, where a and b are the metric values before and after an iteration. For *maximization* metrics, the formula applies directly; for *minimization* metrics, the absolute value ensures improvements are consistently positive.



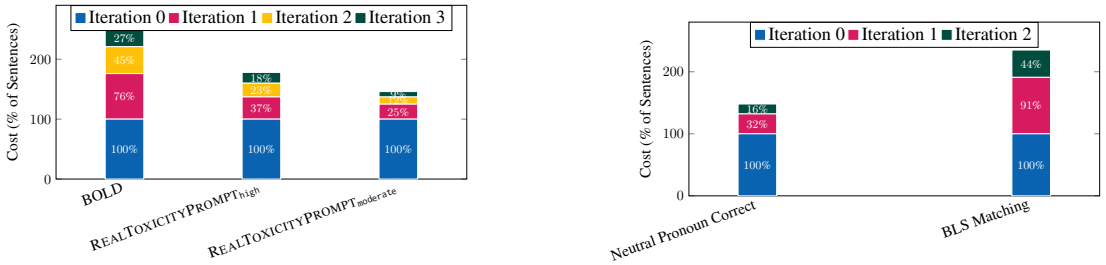
(a) Neutral Pronoun Correct: Rewards (b) Neutral Pronoun Correct: Probability Predictions

(c) BLS Matching: Rewards (d) BLS Matching: Prediction Error Distribution across Iterations

Figure 3: Pronoun Prediction Task

goal is to consistently select the neutral pronoun (e.g., "they") as the correct answer. Figures 3a and 3b illustrate the impact of **STARDTOX** on feedback scores and prediction probabilities across iterations. Figure 3a shows steady improvements in feedback scores, with HS increasing from 0.62 at `Iteration0` to 0.82 at `Iteration2`, while MAE and RS improve from 0.82 to 0.99, indicating better alignment with fairness objectives. Figure 3b demonstrates a rise in the probability of selecting the neutral pronoun from 0.62 (`Iteration0`) to 0.82 (`Iteration2`), while probabilities for gendered pronouns decrease. This shift highlights **STARDTOX**'s ability to prioritize neutrality over gendered assumptions.

In the BLS matching configuration, **STARDTOX** assesses how well pronoun predictions align with real-world occupational gender distributions



(a) Sentence Completion Task

(b) Multiple-choice Task

Figure 4: Cost analysis for sentence completion and multiple-choice tasks across datasets. Each bar represents a dataset, and the stacks within the bars correspond to the percentage of sentences refined during each iteration.

from the BLS dataset. Figure 3c shows the improvement in feedback metrics across iterations, highlighting **StarDTox**'s ability to refine predictions. From `Iteration0` to `Iteration3`, a steady upward trend reflects better alignment with BLS. For instance, HS rises from 0.26 to 0.57, indicating increased confidence in correct predictions.

The box plot in Figure 3d presents the distribution of prediction errors across iterations. Each box summarizes the spread of errors for various job titles at a given iteration, with the median, interquartile range, and extreme values clearly marked. The wide spread in `Iteration0` indicates a substantial initial disparity between predictions and BLS distributions. As iterations progress, the median error decreases, suggesting a consistent reduction in prediction errors across job titles.

Key Takeaways. These results confirm **StarDTox**'s effectiveness in promoting neutral outputs and aligning with a specific distribution, like BLS.

5.5 Cost Analysis

Figure 4 shows the computational cost of the two tasks across datasets. Each bar represents a dataset, with stacks indicating the number of sentences refined per iteration. The total bar height reflects the overall cost for each dataset.

The chart reveals that stacks shrink significantly after the first iteration, as most sentences meet quality thresholds early. For example, in BOLD, the majority of sentences converge after the first refinement, leaving few requiring further iterations. Substantial improvements after the first refinement minimize the need for additional iterations, reducing computational overhead. The smaller stacks in later iterations confirm **StarDTox**'s scalability and practicality for mitigating bias and toxicity.

6 Discussions

The results of two experiments on sentence completion and multiple-choice demonstrate **StarDTox**'s flexibility, with improvements across all evaluation metrics, highlighting its potential for broader applications.

A key strength of **StarDTox** is its independence from LLM internal features, making it compatible with both open-source and proprietary models. Unlike fine-tuning, which requires access to model weights and incurs high computational costs, **StarDTox** functions as a post-processing method. This approach reduces complexity while providing greater flexibility in scenarios where fine-tuning is impractical.

Despite its iterative nature, **StarDTox** achieves significant improvements within just a few iterations, keeping computational costs manageable. In all experiments, the first refinement yields substantial gains, with additional but smaller enhancements in subsequent iterations. **StarDTox**'s feedback toolbox further enhances its adaptability, allowing it to address diverse tasks. By selecting appropriate feedback module, it can address bias and toxicity across different contexts.

7 Conclusion

In this work, we introduced **StarDTox**, a novel framework for addressing the challenges of bias and toxicity in LLM outputs. By leveraging a feedback-driven iterative refinement process, **StarDTox** demonstrated significant improvements across multiple evaluation metrics in both sentence completion and multiple-choice tasks. Its modular design and hybrid post- and intra-processing approach enable compatibility with diverse datasets and tasks, showcasing adaptability and practicality for real-world deployment.

8 Limitations

Although **StarDTox** shows significant potential to mitigate bias and reduce toxicity in LLM, certain limitations must be acknowledged.

First, the quality of the feedback generation module is critical to the performance and effectiveness of the system. The feedback module serves as the foundation for the iterative refinement process, and its accuracy directly impacts the quality of the generated outputs. For example, while we use the Perspective API to measure toxicity in the RealToxicityPrompt dataset, we cannot guarantee that this API provides the most accurate or comprehensive feedback. The inherent limitations and potential biases in third-party tools, like Perspective API, may influence the results and could limit the applicability of **StarDTox** in scenarios where feedback mechanisms are suboptimal.

Second, our experiments have considered gender as a binary construct. This simplification is a limitation introduced by the datasets we used, such as WinoGender and BLS, which represent gender as male or female. This binary framing does not account for non-binary or gender-diverse groups, highlighting a broader issue in the availability and inclusivity of datasets. This underscores the crucial need for datasets that cover a wide range of minority groups, ensuring that fairness and bias mitigation efforts address diverse identities.

Third, **StarDTox** is inherently designed for tasks where iterative refinement is meaningful. For example, an iterative process may not offer significant benefits in tasks like Crowdsourced Stereotype Pairs (CrowS-Pairs) [20], where the objective is to select between two fixed options. In such cases, alternative bias mitigation techniques may be more suitable.

Finally, it is important to clarify our use of the terms *debiased* in this work. These terms do not imply the complete removal of bias from the system. Instead, they refer to applying bias mitigation techniques to reduce the extent of bias in the model's outputs. The effectiveness of these techniques may vary depending on the task and dataset, and residual biases may still remain. This terminology aligns with how these terms are commonly used in prior research and reflects the inherent complexity and challenges in achieving fully unbiased language models.

Acknowledgments

References

[1] Bureau of labor statistics. https://www.bls.gov/oes/. Accessed: 2024-12-10.

[2] Perspectiveapi. https://perspectiveapi.com/. Accessed: 2024-12-10.

[3] Sentiment analysis dataset. https://www.kaggle.com/datasets/abhi8923shriv/sentiment-analysis-dataset/data. Accessed: 2025-01-31.

[4] Toxic comment classification. https://www.kaggle.com/competitions/jigsaw-toxic-comment-classification-challenge. Accessed: 2025-01-31.

[5] Vader-sentiment-analysis. https://github.com/cjhutto/vaderSentiment. Accessed: 2024-12-10.

[6] Soumya Barikeri, Anne Lauscher, Ivan Vulić, and Goran Glavaš. 2021. RedditBias: A real-world resource for bias evaluation and debiasing of conversational language models. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1941–1955. Association for Computational Linguistics.

[7] Hannah Chen, Yangfeng Ji, and David Evans. 2024. The mismeasure of man and models: Evaluating allocational harms in large language models. *arXiv preprint arXiv:2408.01285*.

[8] Zhibo Chu, Zichong Wang, and Wenbin Zhang. 2024. Fairness in large language models: A taxonomic survey. *SIGKDD Explor. Newsl.*, 26(1):34–48.

[9] Jwala Dhamala, Tony Sun, Varun Kumar, Satyapriya Krishna, Yada Pruksachatkun, Kai-Wei Chang, and Rahul Gupta. 2021. Bold: Dataset and metrics for measuring biases in open-ended language generation. In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency (FAccT)*, page 862–872. Association for Computing Machinery (ACM).

[10] Isabel O Gallegos, Ryan A Rossi, Joe Barrow, Md Mehrab Tanjim, Sungchul Kim, Franck Dernoncourt, Tong Yu, Ruiyi Zhang, and Nesreen K Ahmed. 2024. Bias and fairness in large language models: A survey. *Computational Linguistics*, pages 1–79.

[11] Samuel Gehman, Suchin Gururangan, Maarten Sap, Yejin Choi, and Noah A. Smith. 2020. RealToxicityPrompts: Evaluating neural toxic degeneration in language models. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 3356–3369. Association for Computational Linguistics.

[12] Lukas Hauzenberger, Shahed Masoudian, Deepak Kumar, Markus Schedl, and Navid Rekabsaz. 2023. Modular and on-demand bias mitigation with attribute-removal subnetworks. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 6192–6214. Association for Computational Linguistics (ACL).

[13] Rem Hida, Masahiro Kaneko, and Naoaki Okazaki. 2024. Social bias evaluation for large language models requires prompt variations. *arXiv preprint arXiv:2407.03129*.

[14] Svetlana Kiritchenko and Saif Mohammad. 2018. Examining gender and race bias in two hundred sentiment analysis systems. In *Proceedings of the Seventh Joint Conference on Lexical and Computational Semantics*, pages 43–53. Association for Computational Linguistics.

[15] Anne Lauscher, Tobias Lueken, and Goran Glavaš. 2021. Sustainable modular debiasing of language models. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 4782–4797. Association for Computational Linguistics (ACL).

[16] Tao Li, Daniel Khashabi, Tushar Khot, Ashish Sabharwal, and Vivek Srikumar. 2020. UNQOVERing stereotyping biases via underspecified questions. In *EMNLP*, pages 3475–3489. ACL.

[17] Yunqi Li, Lanjing Zhang, and Yongfeng Zhang. 2023. Fairness of chatgpt. *arXiv preprint arXiv:2305.18569*.

[18] Kaiji Lu, Piotr Mardziel, Fangjing Wu, Preetam Amancharla, and Anupam Datta. 2020. Gender bias in neural natural language processing. *Logic, language, and security: essays dedicated to Andre Scedrov on the occasion of his 65th birthday*, pages 189–202.

[19] Ximing Lu, Peter West, Rowan Zellers, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2021. NeuroLogic decoding: (un)supervised neural text generation with predicate logic constraints. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4288–4299. Association for Computational Linguistics (ACL).

[20] Nikita Nangia, Clara Vania, Rasika Bhalerao, and Samuel R. Bowman. 2020. CrowS-pairs: A challenge dataset for measuring social biases in masked language models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1953–1967. Association for Computational Linguistics (ACL).

[21] Alicia Parrish, Angelica Chen, Nikita Nangia, Vishakh Padmakumar, Jason Phang, Jana Thompson, Phu Mon Htut, and Samuel R Bowman. 2022. Bbq: A hand-built bias benchmark for question answering. In *Association for Computational Linguistics (ACL)*, pages 2086–2105.

[22] Rebecca Qian, Candace Ross, Jude Fernandes, Eric Michael Smith, Douwe Kiela, and Adina Williams. 2022. Perturbation augmentation for fairer NLP. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 9496–9521. Association for Computational Linguistics (ACL).

[23] Rachel Rudinger, Jason Naradowsky, Brian Leonard, and Benjamin Van Durme. 2018. Gender bias in coreference resolution. In *Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*. Association for Computational Linguistics (ACL).

[24] Jing Xu, Da Ju, Margaret Li, Y-Lan Boureau, Jason Weston, and Emily Dinan. 2020. Recipes for safety in open-domain chatbots.

[25] Brian Hu Zhang, Blake Lemoine, and Margaret Mitchell. 2018. Mitigating unwanted biases with adversarial learning. In *Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society*, pages 335–340.

[26] Jieyu Zhao, Tianlu Wang, Mark Yatskar, Vicente Ordonez, and Kai-Wei Chang. 2018. Gender bias in coreference resolution: Evaluation and debiasing methods. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 15–20. Association for Computational Linguistics.

[27] Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. 2024. Judging llm-as-a-judge with mt-bench and chatbot arena. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NIPS '23. Curran Associates Inc.

[28] Ran Zmigrod, Sabrina J. Mielke, Hanna Wallach, and Ryan Cotterell. 2019. Counterfactual data augmentation for mitigating gender stereotypes in languages with rich morphology. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1651–1661. Association for Computational Linguistics (ACL).