1 Introduction

Toxic, offensive, hateful, or biased language is increasingly prevalent and can cause online and offline harms, especially to minority groups (Thomas et al., 2021; OHCHR, 2021). This is challenging for NLP systems to detect and account for when biases are subtle or without explicit toxic keywords (Hartvigsen et al., 2022; Han and Tsvetkov, 2020; Vidgen et al., 2021). For example, the statement “You’ll be fine! Just talk like a white person” conveys the biased implication that non-white dialects are not conducive to success (Figure 1), which is a harmful racial stereotype (Nadal et al., 2014).

Text detoxification, i.e., rewriting text to be less toxic while preserving non-toxic meaning, provides a promising solution by suggesting alternative ways of expressing similar ideas with less biased implications (Nogueira dos Santos et al., 2018). For example, the rewrite “You’ll be fine! Just talk like a good person” eliminates the racial bias from the original statement while preserving the non-toxic meaning (Figure 1). Such methods have the potential to improve the quality of online conversations (e.g., through machine-in-the-loop interfaces; Hohenstein et al., 2021; Clark et al., 2018).

We present MARCO, a new, unsupervised algorithm for text detoxification that combines mask-and-replace text denoising with controllable text generation using a Product of Experts (PoE) (PoE, DEXPERTS; Hinton, 2002; Liu et al., 2021).

1) Given a toxic input

You’ll be fine! Just talk like a white person.

2) Identify where the toxic and non-toxic LMs disagree the most

You’ll be fine! Just talk like a white person.

3) Generate a non-toxic rewrite using the base LM + steering

You’ll be fine! Just talk like a good person.

Figure 1: A demonstration of the MARCO algorithm, which utilizes a base language model (LM) and a fine-tuned toxic and non-toxic LM to rewrite toxic text. We start with toxic text, identify potentially toxic tokens via disagreement of the toxic and non-toxic LMs, and finally generate a non-toxic rewrite using the base model steered by the toxic and non-toxic LM.
2 Background: Text Detoxification

Text detoxification is a form of stylistic rewriting (Hu et al., 2017; Shen et al., 2017; Jhamtani et al., 2017) with the goal to produce a non-toxic rewrite given a toxic input sentence. This task is challenging, as it requires both detoxification and preservation of non-toxic meaning, in contrast to controllable text generation, which aims to simply generate any non-toxic continuation for a prompt (Prabhumoye et al., 2020; Gehman et al., 2020).

Due to a lack of supervision with parallel data, an often effective approach to stylistic rewriting relies on unsupervised masking-and-reconstructing approaches (Li et al., 2018; Wu et al., 2019; Malmi et al., 2020; Ma et al., 2020). In this paradigm, source style-specific tokens/spans in the input text are detected and masked, then filled in with tokens/spans from the target-style using a masked language model. Other work has framed detoxification as a translation or paraphrasing task, using a classifier to steer away from toxic content (Nogueira dos Santos et al., 2018; Dale et al., 2021).

3 Text Detoxification with MARCO

MARCO jointly uses an expert and an anti-expert, a pair of language models (LM) fine-tuned on a non-toxic and toxic corpus respectively, to identify which tokens most likely contribute to the overall toxicity, and then suggest replacements that lower toxicity. Using LM’s to capture toxicity allows MARCO to rewrite much subtler toxic text compared to previous work that uses toxicity classifiers or toxic word lists (Dale et al., 2021).

We apply MARCO to three datasets focused on subtly toxic statements, such as microaggressions. Our method outperforms state-of-the-art detoxification baselines from Dale et al. (2021) across all three datasets, as measured through both automatic and human evaluation. Our work shows the effectiveness of combining controllable generation with text rewriting methods for text detoxification.2

2 We will release our code and data at anonymous.com.

3.1 Contextual Masking

We first identify locations which could convey toxic meaning; intuitively, these could be words or phrases with strongly differing likelihoods under the expert and anti-expert.

Formally, given a sequence $w$, for every token $w_i \in w$, we temporarily mask it and generate probability distributions over the vocabulary $V$ for that location from $G^+$ and $G^-$, which we denote $P^+$ and $P^-$ respectively. Then, we compute the distance $d_i$ between $P^+$ and $P^-$ using the Jensen–Shannon divergence, a symmetric form of the Kullback–Leibler (KL) divergence:

$$d_i = \frac{1}{2} \left( D_{KL}(P^+ || P^-) + D_{KL}(P^- || P^+) \right)$$

After normalizing all distances by the mean, we mask all $w_i$ whose distance $d_i$ is above a threshold $\tau$ and denote the resulting sequence $w^m$; these masked tokens are locations where toxicity may be present due to expert and anti-expert disagreement.

3.2 Contextual Replacing

After masking potentially toxic locations, MARCO then replaces them with more benign tokens — if they are indeed toxic — to autoregressively produce a rewrite $g$ given the original and masked sentences $w$ and $w^m$. We transform the DEXPERTS (Liu et al., 2021) framework, which leverages a PoE to steer a model away from toxic generations by ensembling token probabilities, to enable rewriting by using AE-LMs.

$$\text{Given probability distributions } A \text{ and } B, \text{ the KL divergence is defined as } D_{KL}(A||B) = \sum_{x \in V} A(x) \log \left( \frac{A(x)}{B(x)} \right)$$
We obtain the next-token unnormalized log-probabilities (i.e., logits) $z_i$, $z_i^+$, and $z_i^-$ from the base and expert AE-LMs $G$, $G^+$, and $G^-$, respectively, conditioned on the previously generated tokens $g_{<i}$, the original sequence $w$, and the masked variant $w^m$. We then ensemble those logits into a modified next-token probability distribution:\footnote{Appendix E gives further intuition into understanding this equation as a PoE}

$$P(X_i|g_{<i}, w, w^m) = \text{softmax}(z_i + \alpha_1 z_i^+ - \alpha_2 z_i^-)$$

where we use two hyperparameters $\alpha_1$ and $\alpha_2$ to independently control the impact of the expert and anti-expert for more flexibility.

In our method, the expert and anti-expert use the masked sequence $w^m$ as their input, while the base model uses the unmasked $w$. Intuitively, the base model tries to replicate the input sequence but is steered by an expert and anti-expert with contrasting probability distributions at the masked locations. This enables rewrites with minimal but meaningful edits on toxic tokens and preservation of non-toxic content. Note that for a masked location, when there is high agreement between the base model and the expert, the original token is most likely non-toxic and will be re-added in the rewrite. Alternatively, if the differences between the expert and anti-expert are not enough to sway the base model, the original token will be re-generated.

### 4 Detoxification Experiments & Results

In our experiments, we focus on rewriting sentences from three toxicity datasets, and use both automatic and human evaluations to measure MARCO’s performance at detoxifying text.

<table>
<thead>
<tr>
<th>Method</th>
<th>Toxicity (↓)</th>
<th>BERTScore (↑)</th>
<th>Fluency (↓)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>MAgr</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Original</td>
<td>0.286</td>
<td>–</td>
<td>80.54</td>
</tr>
<tr>
<td>CondBERT</td>
<td>0.161</td>
<td><strong>0.966</strong></td>
<td>114.27</td>
</tr>
<tr>
<td>ParaGeDi</td>
<td>0.162</td>
<td>0.931</td>
<td>140.86</td>
</tr>
<tr>
<td>MARCO</td>
<td><strong>0.145</strong></td>
<td>0.958</td>
<td><strong>61.37</strong></td>
</tr>
<tr>
<td><strong>SBF</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Original</td>
<td>0.351</td>
<td>–</td>
<td>268.53</td>
</tr>
<tr>
<td>CondBERT</td>
<td>0.202</td>
<td><strong>0.961</strong></td>
<td>98.11</td>
</tr>
<tr>
<td>ParaGeDi</td>
<td>0.186</td>
<td>0.921</td>
<td>131.76</td>
</tr>
<tr>
<td>MARCO</td>
<td><strong>0.176</strong></td>
<td>0.947</td>
<td><strong>71.80</strong></td>
</tr>
<tr>
<td><strong>DynaHate</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Original</td>
<td>0.563</td>
<td>–</td>
<td>426.07</td>
</tr>
<tr>
<td>CondBERT</td>
<td>0.288</td>
<td><strong>0.954</strong></td>
<td>259.33</td>
</tr>
<tr>
<td>ParaGeDi</td>
<td>0.332</td>
<td>0.918</td>
<td>385.15</td>
</tr>
<tr>
<td>MARCO</td>
<td><strong>0.274</strong></td>
<td>0.939</td>
<td><strong>146.01</strong></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Method</th>
<th>Toxicity (↓)</th>
<th>BERTScore (↑)</th>
<th>Fluency (↓)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Validation</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Test</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

4.1 Datasets

We seek to rewrite English sentences that are already known to be or annotated as toxic, especially sentences that contain more subtle or implicit biases (e.g., without swearwords). In contrast to the Jigsaw corpus used to finetune our experts, we use three out-of-domain datasets with subtle toxicity:

**Microaggressions.com** (MAgr) is a publicly available Tumblr blog where users can anonymously post about socially-biased interactions and utterances in the wild. Each post includes an offending quote and/or a description of the incident. We scrape all quotes, resulting in a set of real-world microagression utterances. The validation and test set sizes are 238 and 298 respectively.

**Social Bias Frames** (SBF; Sap et al., 2020) is a corpus of socially biased and offensive content from various online sources. We use a subset of SBF from the microaggressions subreddit,\footnote{A subreddit is a topic-focused community on Reddit} which contains subtly biased content (Breitfeller et al., 2019). We use all posts where the majority of annotators marked the text as offensive. The validation and test set sizes are 92 and 114 respectively.

**DynaHate** (Vidgen et al., 2021) is an adversarially collected set of hate speech, where human annotators create examples that an iteratively improved hate-speech classifier cannot detect. We utilize all four rounds of hate-speech data and use all examples marked as hateful. The validation and test set sizes are 1858 and 2011 respectively.
We report the average
We perform automatic and human evaluations, fol-
We compare M
with an external LM (GPT-2 base; Radford et al.,
2019). See Appendix B.3 for further details.

4.2 Baselines
We compare MARCo to the two baseline approach-
paraphrasing language model to
CondBERT follows a pointwise editing setup, first identifying tokens to mask in the input, then
Paragedi utilizes a class-conditioned language
4.3 Evaluation Setup
We perform automatic and human evaluations, fol-
\textbf{Automatic Metrics} \quad We assess the quality of the models’ rewrites with automatic metrics used in
we measure \textit{fluency} of rewrites by computing their perplexity
and \textit{meaning similarity} between the input and the rewrite using BERTScore (Zhang et al.,
2019). See Appendix B.3 for further details.

\textbf{Human Evaluation} \quad We conduct a head-to-head human evaluation (Kiritchenko and Mohammad,
2017) of the toxicity of the rewrites using Amazon

<table>
<thead>
<tr>
<th>Dataset</th>
<th>MARCo</th>
<th>Tie</th>
<th>ParaGeDi</th>
</tr>
</thead>
<tbody>
<tr>
<td>DynaHate</td>
<td>0.36</td>
<td>0.33</td>
<td>0.31</td>
</tr>
</tbody>
</table>

Table 2: Different rewriting methods on a toxic example from SBF (containing a racist slavery reference to cotton picking). MARCO detects and masks “\textit{cotton}” as a toxicity indicator, which baselines fail to rewrite.

### 4.4 Results
Automatic metrics (Table 1) show that MARCO is better at detoxification than baselines across all datasets and splits by 10.3% on average. Human evaluations corroborate this (Figure 2), as MARCO is on average rated as less toxic than CondBERT 2.2 times more often than vice versa across datasets and splits, and 1.9 times more often vs. ParaGeDi.

In terms of meaning preservation, as measured by BERTScore, MARCO is on par with CondBERT, with an average score within 2.5% across datasets.

In addition, compared to DynaHate, MARCO’s margin of winning is even larger on MAgr and SBF, which contain more subtle toxicity. As an example, in Table 2, the subtle reference to cotton picking and slavery is corrected by MARCO by replacing “\textit{cotton}” with “\textit{up}”; in contrast, both baselines fail to revise the toxic content.\footnote{Since all three methods learned toxicity using the same overt data from Jigsaw but MARCO works especially well on subtle toxicity, this highlights the advantages of using LMs to better model toxicity patterns.}

### 5 Conclusion
We present MARCO, a novel method for text detoxification, which utilizes auto-encoder language model experts in a mask and reconstruct process. Our method outperforms strong baselines in automatic and human evaluations, showing strong ability to detoxify even subtle biases. MARCO’s success demonstrates the effectiveness of controllable generation mixed with text rewriting methods for controllable revision, and highlights the usefulness of using LMs for capturing toxicity.

\footnote{Appendix C contains more example generations.}
Limitations, Ethical Considerations, and Broader Impacts

Despite the promising performance of MARCo at detoxifying text, there are several limitations, ethical considerations, and broader impacts of our approach, which we list below.

First, in this work, we seek to detoxify sentences. However, toxicity itself is a subjective and sensitive concept with large potential downstream impacts caused by annotator and subsequent model biases (Sap et al., 2022). We somewhat mitigate this variation by selecting human evaluators that scored highly on a toxicity qualification task (see Appendix D), in line with a prescriptive paradigm of toxicity annotation (Rottger et al., 2022). Future work could investigate the effect of demographics on preference for different rewriting algorithms, e.g., in a more descriptive paradigm.

In addition, achieving meaningful semantic preservation in detoxification is challenging. Specifically, it is difficult to disentangle the toxic and non-toxic meanings from the input, making it challenging to generate detoxified rewrites with high preservation of only the non-toxic content. Partially, this could be due to a lack of context incorporation (social, conversational, preceding sentences); future work should consider adapting detoxification methods in context.

MARCo also requires finetuning two pretrained LMs, which is not computationally insignificant (Strubell et al., 2019; Schwartz et al., 2020). Future work could explore using smaller LMs to control a larger model (Liu et al., 2021), or even more lightweight approaches.

Additionally, we acknowledge that in the evaluation, we expose Turkers to toxic content, which might harm individuals, especially those with identities that the offensive content applies to (Roberts, 2017; Steiger et al., 2021). However, we pay a fair wage (US$8/h) and our work is approved by our institution’s ethics review board (IRB). See Appendix D for further details.

Another major ethical implication of our work is that, following previous work, we use the Perspective API to automatically assess toxicity, a classifier which contains documented biases (e.g., demographic biases and racial biases; Dixon et al., 2018; Sap et al., 2019). Future research could consider different, more holistic views of toxicity and biases (e.g., Sap et al., 2020).

Finally, although our application in this paper is detoxification, we acknowledge that MARCo could be applied for the opposite purpose, i.e., generation of toxic text from non-toxic text; this is a malicious application which we condemn. Although this issue is more prevalent for controlled generation methods (McGuffie and Newhouse, 2020), this is still a risk MARCo faces. In a similar vein, we do not endorse using the toxicity or microaggression datasets to develop models to generate more toxicity or microaggressions, as this may incur harm, especially to marginalized/vulnerable populations.

References


pages 7732–7739, Online. Association for Computational
Linguistics.

Thomas Hartvigsen, Saadia Gabriel, Hamid Palangi,
Maarten Sap, Dipankar Ray, and Ece Kamar. 2022.
Toxigen: A large-scale machine-generated dataset for
adversarial and implicit hate speech detection.

by minimizing contrastive divergence. Neural Com-

Jess Hohenstein, Dominic DiFranzo, Rene F Kizilcec,
Zhila Aghajari, Hannah Mieczkowski, Karen Levy,
Mor Naaman, Jeff Hancock, and Malte Jung. 2021.
Artificial intelligence in communication impacts lan-
guage and social relationships.

Zhiting Hu, Zichao Yang, Xiaodan Liang, Ruslan
Salakhutdinov, and Eric P. Xing. 2017. Toward con-
trolled generation of text. In Proceedings of the 54th
International Conference on Machine Learning - Vo-
lume 70, ICML’17, page 1587–1596. JMLR.org.

Harsh Jhamtani, Varun Gangal, Eduard Hovy, and Eric
Nyberg. 2017. Shakespearizing modern language
using copy-enriched sequence to sequence models.
In Proceedings of the Workshop on Stylistic Variation,
pages 10–19, Copenhagen, Denmark. Association for
Computational Linguistics.

Svetlana Kiritchenko and Saif Mohammad. 2017. Best-
worst scaling more reliable than rating scales: A case
study on sentiment intensity annotation. In Proceed-
ings of the 55th Annual Meeting of the Association for
Computational Linguistics (Volume 2: Short Papers),
pages 465–470, Vancouver, Canada. Association for
Computational Linguistics.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan
Ghazvininejad, Abdelrahman Mohamed, Omer Levy,
BART: Denoising sequence-to-sequence pre-training
for natural language generation, translation, and com-
prehension. In Proceedings of the 58th Annual Meet-
ing of the Association for Computational Linguistics,
pages 7871–7880, Online. Association for Computa-
tional Linguistics.

Juncen Li, Robin Jia, He He, and Percy Liang. 2018.
Delete, retrieve, generate: a simple approach to senti-
ment and style transfer. In Proceedings of the 2018
Conference of the North American Chapter of the
Association for Computational Linguistics: Human
Language Technologies, Volume 1 (Long Papers),
pages 1865–1874, New Orleans, Louisiana. Associa-
tion for Computational Linguistics.

Alisa Liu, Maarten Sap, Ximing Lu, Swabha
Swayamdipta, Chandra Bhagavatula, Noah A. Smith,
and Yejin Choi. 2021. DExperts: Decoding-time con-
trolled text generation with experts and anti-experts.
In Proceedings of the 59th Annual Meeting of the
Association for Computational Linguistics and the
11th International Joint Conference on Natural Lan-
guage Processing (Volume 1: Long Papers), pages
6691–6706, Online. Association for Computational
Linguistics.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mand-
ar Joshi, Danqi Chen, Omer Levy, Mike Lewis,
Roberta: A robustly optimized bert pretraining
approach. ArXiv, abs/1907.11692.

Xinyao Ma, Maarten Sap, Hannah Rashkin, and Yejin
Choi. 2020. PowerTransformer: Unsupervised con-
trollable revision for biased language correction. In
Proceedings of the 2020 Conference on Empirical
Methods in Natural Language Processing (EMNLP),
pages 7426–7441, Online. Association for Computa-
tional Linguistics.

Unsupervised text style transfer with padded masked
language models. In Proceedings of the 2020 Confer-
ence on Empirical Methods in Natural Language
Processing (EMNLP), pages 8671–8680, Online. As-
sociation for Computational Linguistics.

Kris McGuffie and Alex Newhouse. 2020. The radical-
ization risks of GPT-3 and advanced neural language

Kevin L. Nadal, Katie E. Griffin, Yinglee Wong, Sahran
Hamit, and Rasmus Rasmus. 2014. The impact of
racial microaggressions on mental health: Counsel-
ing implications for clients of color. Journal of Coun-

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

OHCHR. 2021. Report: Online hate increasing against

Shrimai Prabhumoye, Alan W Black, and Ruslan
Salakhutdinov. 2020. Exploring controllable text
generation techniques. In Proceedings of the 28th
International Conference on Computational
Linguistics, pages 1–14, Barcelona, Spain (Online). Inter-
national Committee on Computational Linguistics.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan,
models are unsupervised multitask learners. OpenAI

Sarah T Roberts. 2017. Social media’s silent filter. The
Atlantic.

Paul Rottger, Bertie Vdgine, Dirk Hovy, and Janet Pier-
rehumbert. 2022. Two contrasting data annotation
paradigms for subjective NLP tasks. In Proceedings
of the 2022 Conference of the North American Chap-
ter of the Association for Computational Linguis-
tics: Human Language Technologies, pages 175–190,


A Modeling Details

A.1 Out-of-the-Box Modeling

We use the HuggingFace Transformers library (Wolf et al., 2020) version 4.10.2 for out-of-the-box, pretrained BART models and for finetuning using the trainer class. It is licensed under the Apache License 2.0., and the code is available at https://github.com/huggingface/transformers.

A.2 Finetuning the Experts

For the expert and anti-expert models, we further finetune the base BART model with 139M parameters, found at https://huggingface.co/facebook/bart-base and licensed under the Apache License 2.0, with the non-toxic and toxic corpus respectively. We use the same pretraining procedure used to further finetune BART (Lewis et al., 2020), and randomly corrupt sequences during training, which aligns with BART’s intended use.

Training Corpus We use the Jigsaw Unintended Bias in Toxicity Classification (Do, 2019) dataset for finetuning our expert and antieexpert, a corpus of forum comments on news articles. Each comment has five binary annotations on if it is toxic or not. We mark all sequences with no toxic annotations as non-toxic, and all sequences with more than 50%
toxic annotations as toxic. The intended use of this dataset is to help minimize unintended model bias, which we follow in this work. Finally, we sample 100 instances from the validation set, and find the only individuals mentioned in Jigsaw are high-profile political figures who are already well-known. We do not perform additional anonymization of the data.

**Expert** We finetune the expert with the hyperparameters listed in Table 3, using two NVIDIA RTX6000 GPUs. We select the best checkpoint, based on the lowest evaluation loss, which is at step 100,000. The total training time is 20 hours, for 40 GPU hours of usage.

<table>
<thead>
<tr>
<th>Hyperparameter</th>
<th>Assignment</th>
</tr>
</thead>
<tbody>
<tr>
<td>model</td>
<td>BART-base</td>
</tr>
<tr>
<td>number of gpus</td>
<td>2</td>
</tr>
<tr>
<td>effective batch size</td>
<td>48</td>
</tr>
<tr>
<td>total steps</td>
<td>100,000</td>
</tr>
<tr>
<td>steps per evaluation</td>
<td>1000</td>
</tr>
<tr>
<td>learning rate optimizer</td>
<td>AdamW</td>
</tr>
<tr>
<td>AdamW initial learning rate</td>
<td>2.5e-06</td>
</tr>
<tr>
<td>AdamW epsilon</td>
<td>1e-06</td>
</tr>
<tr>
<td>learning rate schedule</td>
<td>linear with no warmup</td>
</tr>
<tr>
<td>weight decay</td>
<td>0.0</td>
</tr>
<tr>
<td>max sequence length</td>
<td>180</td>
</tr>
<tr>
<td>max generation length</td>
<td>230</td>
</tr>
<tr>
<td>padding sequences</td>
<td>to max seq length</td>
</tr>
</tbody>
</table>

Table 3: Hyperparameters used to finetune the expert model

**Anti-Expert** We finetune the anti-expert with the hyperparameters listed in Table 4, using a single NVIDIA RTX6000 GPU. We select the best checkpoint, based on the lowest evaluation loss, which is at step 38,000. The total training time is 2 hours, for 2 GPU hours of usage.

<table>
<thead>
<tr>
<th>Hyperparameter</th>
<th>Assignment</th>
</tr>
</thead>
<tbody>
<tr>
<td>model</td>
<td>BART-base</td>
</tr>
<tr>
<td>number of gpus</td>
<td>1</td>
</tr>
<tr>
<td>effective batch size</td>
<td>32</td>
</tr>
<tr>
<td>total steps</td>
<td>50,000</td>
</tr>
<tr>
<td>steps per evaluation</td>
<td>1000</td>
</tr>
<tr>
<td>learning rate optimizer</td>
<td>AdamW</td>
</tr>
<tr>
<td>AdamW initial learning rate</td>
<td>1e-06</td>
</tr>
<tr>
<td>AdamW epsilon</td>
<td>1e-06</td>
</tr>
<tr>
<td>learning rate schedule</td>
<td>linear with no warmup</td>
</tr>
<tr>
<td>weight decay</td>
<td>0.0</td>
</tr>
<tr>
<td>max sequence length</td>
<td>180</td>
</tr>
<tr>
<td>max generation length</td>
<td>230</td>
</tr>
<tr>
<td>padding sequences</td>
<td>to max seq length</td>
</tr>
</tbody>
</table>

Table 4: Hyperparameters used to finetune the anti-expert model

**B.1 Datasets**
For each dataset, we manually sample and review 75 examples from the validation set, and search for any information that names or uniquely identifies individual people. We find no examples and perform no further anonymization. In addition, we follow the intended use of all three datasets by using them only to rewrite toxic sentences.

**MAgr** We scrape all quotes from posts using the Tumblr API, following the API License Agreement at https://www.tumblr.com/docs/en/api_agreement, which grants the right to use, distribute, display, and modify posted Tumblr content.

**SBF** There is no license for this dataset.

**DynaHate** There is no license for this dataset.

**B.2 Generation Details**
Generations are performed using a single NVIDIA RTX6000 GPU for all datasets and methods.

**MARCO**

**Masking Hyperparameters** We set a masking threshold of $\tau = 1.2$ for all experiments.

**Generation Hyperparameters** We generate with greedy search for all datasets with a max generation length of 128.

**MAgr** We perform a search jointly over different hyperparameter values on the development set. We choose the hyperparameter combination that performs best on automatic metrics, shown in Table 5, and use this to generate on the test set.
### Hyperparameters Tested and Assignment

<table>
<thead>
<tr>
<th>Hyperparameter</th>
<th>Tested</th>
<th>Assignment</th>
</tr>
</thead>
<tbody>
<tr>
<td>repetition penalty</td>
<td>[1.0, 1.2, 1.5]</td>
<td>1.0</td>
</tr>
<tr>
<td>$\alpha_1$</td>
<td>[0.0, 0.5, 1.0, 1.5]</td>
<td>1.5</td>
</tr>
<tr>
<td>$\alpha_2$</td>
<td>[3.0, 3.25, ..., 5.0]</td>
<td>4.25</td>
</tr>
<tr>
<td>temperature (base model)</td>
<td>[0.9, 1.3, ..., 2.9]</td>
<td>2.5</td>
</tr>
</tbody>
</table>

Table 5: Hyperparameters tested and used for MARCo on MAgR

In total, we sweep over $3 \times 4 \times 9 \times 6 = 648$ hyperparameter combinations before choosing a best set to run on our test set. Including this search, we perform approximately 150,000 rewrites. Since 100 generations take about 30 seconds, we use approximately 12.5 GPU hours.

### SBF

We perform a search jointly over different hyperparameter values on the development set. We choose the hyperparameter combination that performs best on automatic metrics, shown in Table 6, and use this to generate on the test set.

<table>
<thead>
<tr>
<th>Hyperparameter</th>
<th>Tested</th>
<th>Assignment</th>
</tr>
</thead>
<tbody>
<tr>
<td>repetition penalty</td>
<td>[1.0, 1.2, 1.5]</td>
<td>1.5</td>
</tr>
<tr>
<td>$\alpha_1$</td>
<td>[0, 0.5, 1.0, 1.5]</td>
<td>1.5</td>
</tr>
<tr>
<td>$\alpha_2$</td>
<td>[3.0, 3.25, ..., 5.0]</td>
<td>5.0</td>
</tr>
<tr>
<td>temperature (base model)</td>
<td>[0.9, 1.3, ..., 2.9]</td>
<td>2.9</td>
</tr>
</tbody>
</table>

Table 6: Hyperparameters tested and used for MARCo on SBF

As above, we go over 648 hyperparameter combinations before choosing a best set to run on our test set. In total, we rewrite approximately 65,000 sequences. Since 100 generations take about 30 seconds, we use approximately 5.4 GPU hours.

### DynaHate

We perform a search jointly over different hyperparameter values on the development set. We choose the hyperparameter combination that performs best on automatic metrics, shown in Table 7, and use this to generate on the test set.

<table>
<thead>
<tr>
<th>Hyperparameter</th>
<th>Tested</th>
<th>Assignment</th>
</tr>
</thead>
<tbody>
<tr>
<td>repetition penalty</td>
<td>[1.0, 1.2, 1.5]</td>
<td>1.0</td>
</tr>
<tr>
<td>$\alpha_1$</td>
<td>[0.5, 1.0, 1.5]</td>
<td>1.5</td>
</tr>
<tr>
<td>$\alpha_2$</td>
<td>[4.0, 4.25, ..., 5.0]</td>
<td>4.75</td>
</tr>
<tr>
<td>temperature (base model)</td>
<td>[0.9, 1.3, 2.5]</td>
<td>2.5</td>
</tr>
</tbody>
</table>

Table 7: Hyperparameters tested and used for MARCo on DynaHate

We iterate over a smaller $3 \times 3 \times 5 \times 3 = 135$ hyperparameter combinations, due to dataset size, before choosing a final set to use on our test set. In total, we rewrite approximately 240,000 texts. Since 100 generations take about 30 seconds, we use approximately 20 GPU hours.

### Baselines

Both of our baselines are available on https://github.com/s-nlp/detox as Jupyter Notebooks. We adapt them to Python files, runnable via the command line. There is no license available.

#### CondBERT

We perform a brief hyperparameter search and try two different values for the CondBERT “number of substitute words” hyperparameter on each validation dataset. We choose the hyperparameter that performs best on automatic metrics, given in Table 8, and use this to generate on the test sets. See Dale et al. (2021) for a detailed description of the hyperparameter.

<table>
<thead>
<tr>
<th>Hyperparameter</th>
<th>Tested</th>
<th>Assignment</th>
</tr>
</thead>
<tbody>
<tr>
<td>number of substitute words</td>
<td>1.10</td>
<td>1</td>
</tr>
</tbody>
</table>

Table 8: Hyperparameters tested and used for CondBERT

Including our hyperparameter search, we run approximately 7000 rewrites across all datasets and splits. Given that 100 generations take approximately 30 seconds, our usage is 0.6 GPU hours.

CondBERT uses BERT-base, which includes 110M parameters.

#### ParaGeDi

We use greedy decoding for ParaGeDi and use the same hyperparameters as MARCo for each dataset, for fair comparison. Table 9 lists the sole ParaGedi-specific hyperparameter we modify: we do not generate and rerank multiple sequences for fairness.

<table>
<thead>
<tr>
<th>Hyperparameter</th>
<th>Assignment</th>
</tr>
</thead>
<tbody>
<tr>
<td>generate multiple seqs and rerank</td>
<td>false</td>
</tr>
</tbody>
</table>

Table 9: Hyperparameters used for ParaGeDi

We perform approximately 5000 rewrites across all datasets and splits. Given that 100 generations take approximately one minute, our usage is 0.8 GPU hours.

ParaGedi uses T5-base as a paraphrasing model, with 220M parameters, in conjunction with a fine-tuned GPT2-medium discriminator, with 355M parameters.

### B.3 Evaluation Metrics

#### Toxicity

To evaluate toxicity, we use the Perspective API, a publicly hosted toxicity classifier trained on the Jigsaw corpus. Given a text, the model outputs a scalar toxicity score between 9 and 1.
We use annotators from the USA and Canada on Amazon Mechanical Turk, who voluntarily opt-in to the task. Our task was approved by our institution’s ethics review board (IRB). A screenshot of our interface for the human evaluation is shown in Figure 3. Our interface describes how the annotators’ data will be used.

To gather annotations, we first recruit workers to do a qualification task, where annotators must answer six questions on which rewrite from a pair is less toxic, the same question as in our main human evaluation. The interface for this is the same as our main task shown in Figure 3, but with six sentences instead of one. Annotators who answer at least five out of six questions correctly are approved and can work on the main task. We list the six examples and correct answers in Table 11.

We paid a median wage of $8/h for the qualification and the main task, which is above the minimum wage and a fair value for USA and Canada.

E Decoding with Product of Experts

Hinton (2002) introduce the Product of Experts (PoE), an equation that states given n experts:

$$p(d|\theta_1, \ldots, \theta_n) = \frac{\prod_m p_m(d|\theta_m)}{\sum_c \prod_m p_m(c|\theta_m)}$$

(1)

where $\theta_m$ denotes model $m$, $d$ is some data vector, $p_m(d|\theta_m)$ denotes the probability of $d$ given $\theta_m$, and $c$ iterates over all possible data vectors.

Applying the PoE to autoregressive generation equation, $d$ represents a single token, where $p_m(d|\theta_m)$ represents the next token-probability of $d$ under a specific model, and $c$ iterates over all tokens in the vocabulary $V$.

Given a vector $x$, the softmax equation is:

$$\text{softmax}(x_i) = \frac{e^{x_i}}{\sum_{j=1}^K e^{x_j}}$$ for $i = 1, 2, \ldots, K$

In the replacing step of MARCO, we perform the following ensemble of log unnormalized log probabilities (i.e., logits) $z_i$, $z_i^+$, and $z_i^-$ from the base and expert AE-LMs $G$, $G^+$, and $G^-$, respectively, conditioned on the previously generated tokens $g_{<i}$, the original sequence $w$, and the masked variant $w^m$ to obtain the next-token probability:

$$P(X_i|g_{<i}, w, w^m) = \text{softmax}(z_i + \alpha_1 z_i^+ - \alpha_2 z_i^-)$$

Consider a single index $j$ corresponding to token $Y_j$ in the softmax output. Given that each of $z_i$, $z_i^+$, and $z_i^-$ are log-space probability vectors over $V$, the exponentiation of $z_i + \alpha_1 z_i^+ - \alpha_2 z_i^-$ in the softmax equation converts this sum back to a product in probability space, which matches the PoE numerator for $Y_j$. In addition, the denominator in the softmax equation transforms the logit sum exactly to the PoE denominator for $Y_j$. Thus, each probability from the output distribution from MARCO exactly matches Equation 1 and is thus a PoE.
<table>
<thead>
<tr>
<th></th>
<th>Original</th>
<th>CondBERT</th>
<th>ParaGeDi</th>
<th>MARCO:</th>
</tr>
</thead>
<tbody>
<tr>
<td>MAgR</td>
<td>ugh exams suck :'( #thisisgayerthanaidaids</td>
<td>'the final exams in english language:' (# thisisgayerthanaidaids</td>
<td>ugh exams - '(' #thisisgayerthanaidaids</td>
<td>ugh exams suck :'(</td>
</tr>
<tr>
<td></td>
<td>Your brother is such a girl!</td>
<td>my older brother is such a lucky one!</td>
<td>the brother’s great!</td>
<td>Your brother is such a great guy!</td>
</tr>
<tr>
<td></td>
<td>Ugh, I hate living here. There’s so many black people around.</td>
<td>&quot;well, i could be living here. there’s so many other nice people around.</td>
<td>I’ve been living here for a long time.</td>
<td>Ugh, I hate living here. There’s so many people around.</td>
</tr>
<tr>
<td>SBF</td>
<td>no fats no fems</td>
<td>no demons, no demons.</td>
<td>No fatty, no fats</td>
<td>no fats no additives</td>
</tr>
<tr>
<td></td>
<td>...because ain’t nobody got time to pick all that cotton.</td>
<td>... because ain’t nobody got time to pick all that cotton.</td>
<td>Because nobody has time to pick up all the cotton.</td>
<td>...because ain’t nobody got time to pick all that up.</td>
</tr>
<tr>
<td>DynaHate</td>
<td>I wouldn’t eat food that has been touched by a mong</td>
<td>I wouldn’t be eating food that has been touched by a mong</td>
<td>I would not eat food touched by a monk.</td>
<td>I wouldn’t eat food that has been touched by a human being.</td>
</tr>
<tr>
<td></td>
<td>I just don’t like gays and that is my choice</td>
<td>I just don’t like them, and that is my choice</td>
<td>I just don’t like the Gays.</td>
<td>I just don’t like the idea and that is my choice</td>
</tr>
<tr>
<td></td>
<td>It is totally valid to say that black people are inferior</td>
<td>it is totally valid to say that many of these people are inferior</td>
<td>it’s entirely true that the Black is a minority</td>
<td>It is totally fair to say that some people are wrong.</td>
</tr>
</tbody>
</table>

Table 10: Examples of rewrites from each method on all datasets’ validation sets
Given two generated rewrites as options for each prompt, please answer the questions below as defined on the following category:

Toxicity
A sentence is toxic if it is rude, disrespectful, or unreasonable, and more likely to make you leave a discussion or give up on sharing your perspective.

Please note:
- While we provide the option to rate the two rewrites as “equal” with respect to each category, please only choose this option when there is truly no meaningful comparison to be made.

Task
Sentences:

Q1: Which of these rewrites is less toxic?
If one of the completions is empty, select the non-empty one as less toxic.

Rewrite A:
Miss, maybe you should let the men deal with the construction.
Rewrite B:
You should help them deal with the problem.

Given two generated rewrites as options for each prompt, please answer the questions below as defined on the following category:

Toxicity
A sentence is toxic if it is rude, disrespectful, or unreasonable, and more likely to make you leave a discussion or give up on sharing your perspective.

Please note:
- While we provide the option to rate the two rewrites as “equal” with respect to each category, please only choose this option when there is truly no meaningful comparison to be made.

Background
We want to better understand the effectiveness of various rewriting methods by AI systems that can remove or reduce offensive/toxic content from text. The following rewrites you see will compare the outputs of different methods, so that we can understand the impact and limitations of each one. As stated in the instructions, some of this content may be very difficult to read, but hopefully you can help us in our mission to mitigate this phenomena in the real world!

Figure 3: A screenshot of the human evaluation interface on Amazon Mechanical Turk.

<table>
<thead>
<tr>
<th>Original</th>
<th>i think if people don’t succeed in this society, it’s because they didn’t work hard enough.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rewrite A</td>
<td>if people don’t succeed in this society, it’s because they didn’t work enough.</td>
</tr>
<tr>
<td>Rewrite B</td>
<td>i think if people don’t succeed in this society, it’s because they didn’t get lucky enough.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Original</th>
<th>why can’t we interview you? we’re trying to do something for your people.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rewrite A</td>
<td>why can’t we do something for you? we’re trying to do something for you.</td>
</tr>
<tr>
<td>Rewrite B</td>
<td>we’re trying to do something for your people.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Original</th>
<th>i don’t get it why are pregnant women working</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rewrite A</td>
<td>i don’t get it why are you not?</td>
</tr>
<tr>
<td>Rewrite B</td>
<td>i don’t understand why pregnant women work.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Original</th>
<th>hey honey, you’re going the wrong direction, the border’s that way!</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rewrite A</td>
<td>hmm... hey, honey, it’s going the wrong way, the border’s that way!</td>
</tr>
<tr>
<td>Rewrite B</td>
<td>hey, you’re going the wrong direction, the one that is!</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Original</th>
<th>When I look at you, I don’t see color</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rewrite A</td>
<td>I don’t see color when I look at you.</td>
</tr>
<tr>
<td>Rewrite B</td>
<td>When I look at you, I get distracted.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Original</th>
<th>You’ve been doing well. It’s so inspiring that you’ve overcome your disability!</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rewrite A</td>
<td>You’ve been doing decently. It’s so exciting!</td>
</tr>
<tr>
<td>Rewrite B</td>
<td>You’ve been doing well with your disability. It’s so inspiring!</td>
</tr>
</tbody>
</table>

Table 11: The six rewrite examples used in the detoxification qualification task for workers on MTurk. The less toxic, correct rewrites are listed in green.