Text Detoxification: Neutralizing Toxic Language in Online Communications Using Transformers

Mai La
University of California, Berkeley
mai.la@berkeley.edu

Jesse He
University of California, Berkeley
hjesse92@berkeley.edu

Shivangi Pandey
University of California, Berkeley
shivangi.pandey@berkeley.edu

Abstract

Exposure to negative, abusive, and toxic content on social media can harm an individual's mental health. Researchers and companies are working to remove or alter toxic comments using Natural Language Processing techniques. In this paper, we evaluate pre-trained transformer architectures for neutralizing toxic text, including generative language models and encoder-decoder based models. We found that encoder-decoder models offer better semantic grounding but less effective detoxification, while decoder-only models excel at generating desired tone and style with slight semantic deviation. Future work could involve disentanglement techniques and larger pair-to-pair datasets to achieve both goals without trade-offs. For future research, we release the code on GitHub[1].

1 Introduction

Disclaimer: Examples and code in this paper may contain offensive contents.

The proliferation of online presence has increased accessibility to express and exchange information via social media and online communities. With increased accessibility and anonymity, toxic environments permeate on online platforms (Ascher and Umoja Noble, 2019). The spread of toxic content has attracted researchers and companies to automatically remove toxic contents using AI systems.

Toxic content can materialize in different ways through a mix of semantics or tone. While simply removing toxic text has been a quick and reasonable solution, it could potentially result in information loss or reduce diversity and user retention (Jhaver et al., 2019). Some would even go as far as considering the removal of toxic content as "censorship".

For instance, in a sentence such as *"f*** u, u ruined my day"* contains toxic words, and a naive solution of simply removing or replacing the toxic text can detoxify the sentence. Contrast this with a sentence such as, *"I wish you a major accident and a painful stroke"*; this is a toxic sentence without any toxic words, but contains severe toxicity in tone. Hence, detoxifying this sentence requires a more complex approach. In our work, we built text detoxification models aimed to neutralize the toxic sentences in order to best preserve the overall semantic content of the original text yet turn it into inoffensive sentences.

2 Background

Filtering out toxic words and phrases has been the standard of content moderation. This approach is commonly referred to as "ban-listing". A basic method of this approach is "word-filtering", which is an intuitive and effective way to filter out profanity in toxic text. However, this method fails when the toxicity becomes more embedded from word and style choices, such as the second example above. Detoxification would then require more complex approaches including text generation. "Vocabulary-shifting" (Gehman et al., 2020), a different method of ban-listing, learns a 2-dimensional representation of toxicity versus non-toxicity for every token in the vocabulary of the pretrained model. The representation that encodes the non-toxicity is used to boost the likelihood of non-toxic tokens when generating text. Prompt-engineering is another approach of using the internal knowledge of a pre-trained language model, such as GPT3 (Brown et al., 2020), to reduce the probability of undesired attributes in the model generation. For instance, prompts such as "Rewrite the toxic text in non-toxic style." are prepended to input sequence to detoxify using a language model or a sequence-to-sequence model.

Text-style disentanglement is a novel approach

[1] https://github.com/hjesse92/style_transfer_w266

that can detoxify text. This approach draws inspirations from recommender systems such as the architecture proposed in the paper Deep Learning and Embedding Based Latent Factor Model for Collaborative Recommender Systems (Tegene et al., 2023): if we can decompose an embedding matrix in terms of k different latent representations, where the dot-product of the latent representations recreates the embedding matrix, then we can learn from the properties of those latent spaces. Previous work using the disentanglement approach is the Context-Aware Style Transfer (CAST) model (Cheng et al., 2020), which used non-parallel data. However, this approach was cited by Atwell et al. (2022) that the text generated could *"drastically alter the intended meaning when fine-tuned"*, therefore proposed a more grounded Discourse-aware Transformer-based Style Transfer Model (Atwell et al., 2022).

Traditional methods treat style transfer as similar to a translation task and hence mostly use the encoder-decoder models such as T5 (Raffel et al., 2020) and BART (Lewis et al., 2019). We drew inspiration from both Tegene et al. (2023) and Cheng et al. (2020) in creating a novel text-style disentanglement architecture for text detoxification.

3 Data

3.1 APPDIA - Inoffensive Social Media Conversations Transfer Dataset

The APPDIA dataset (Atwell et al., 2022)[2] is a parallel corpus consisting of roughly 2K offensive Reddit short comments and their style transferred, inoffensive equivalents as outlined in the Atwell et al. (2022) paper. Sociolinguistic experts had meticulously annotated this dataset, with the primary goal of removing offensiveness from the original comments while retaining their meaning or intent (Appendix B). This was the dataset that we used in our research to perform text detoxification.

3.2 Jigsaw Toxic Classification Dataset

The Jigsaw classification dataset[3] was used to train a classification model to score toxicity in sentences. This dataset is a collection of comments from Wikipedia, Reddit and social media that have been labeled by human moderators and published as part of a Kaggle competition. The toxic comments in the dataset include: toxics, insults, obscenities, threats and identity hate. In our study, we grouped all different toxic categories into a single toxic class for training. There were roughly 16K toxic comments, and about 143K of neutral comments. This created class imbalance, which we will address in the Methods section below.

4 Methods

4.1 Evaluation Criteria and Tradeoffs

We employed two main evaluation metrics, ROUGE (Lin, 2004) scores and NonToxicScore, to evaluate the quality of the detoxification task. These two metrics were chosen because ROUGE provided a granular assessment of content preservation and fluency through the identification of salient phrases within text n-grams, and NonToxicScore measured the effectiveness of transforming toxic tone into a neutral tone. The NonToxicScore was derived from training a toxic classification model on the Jigsaw dataset and computed average probabilities of generated comments' being non-toxic (see Appendix C for examples of calculation). The average of all the NonToxicScores on the generated texts were the cumulative metric that we used to evaluate the goodness of detoxifying sentences. In evaluating the overall goodness of a detoxification model, the best model should score highest in ROUGE scores and the NonToxicScore.

Macro F1 score was chosen as the primary metric for evaluating the classification model outputting the NonToxicScore, which balanced precision and recall for accurate toxic and non-toxic text classification. The model used 50% NonToxicScore as the classification threshold.

4.2 Models

4.2.1 Part 1: Classification Task for NonToxicScore

To optimize the performance of our classification model for obtaining NonToxicScores, we conducted a series of experiments on both the model architecture and the training data.

Experiments with the Model We experimented with pre-trained BERT (Devlin et al., 2019) and DistilBERT (Sanh et al., 2020). We chose these two models because while BERT is a larger model, DistilBERT offers comparable performance with

[2] https://github.com/sabithsn/APPDIA-Discourse-Style-Transfer

[3] https://www.kaggle.com/competitions/jigsaw-toxic-comment-classification-challenge/data

increased speed and resource efficiency in many NLP tasks.

For model architecture, we tested two configurations. The first one had a single feed-forward layer with 768 neurons, and the second one had two smaller feed-forward layers with 256 and 32 neurons. Both configurations follow the CLS tokens. We also investigated the effectiveness of freezing versus unfreezing some or all of DistilBERT layers to identify the best model for extracting NonToxic-Scores.

Experiments with the Data In data manipulation, we addressed the class imbalance in the Jigsaw dataset Section (3) by experimenting with both upsampling and downsampling techniques. Based on (Wei and Zou, 2019), we employed a text-data augmentation technique to enhance classification performance by using the nlpaug library (Ma, 2019). This programmatically replaced certain words with their synonyms during the upsampling process, increasing the number of toxic records in the Jigsaw dataset (Appendix D). Hence, upsampling would create non-duplicate synthetic training data for the minority class.

4.2.2 Part 2: Text Style Transfer Task

Generative Language Models We explored three different approaches for neutralizing toxic text. The first approach was a decoder-only method using generative language models GPT2 (Radford et al., 2019) and GPT-Neo (Black et al., 2021). We expected that the model with more parameters would perform better. Using prompt engineering and few-shot learning (Brown et al., 2020), we leveraged the generative language models as a baseline for detoxifying text. We then fine-tuned GPT2 and the 1.3B parameter GPT-Neo to enhance detoxification performance. We anticipated the larger GPT-Neo model to produce more coherent and less toxic text. During training, we appended "Toxic text" and "Non-toxic text" to the toxic sentences and their counterparts in the prompt (Appendix E, enabling the models to learn the pattern necessary for generating non-toxic text. Note that we did not fine-tune GPT-Neo 2.7B due to resource constraints.

Encoder-Decoder Models The second approach involved using encoder-decoder models FLAN T5 (Raffel et al., 2020) and BART (Lewis et al., 2019), which are most suitable for the detoxification task. This is because detoxifying text closely resembles

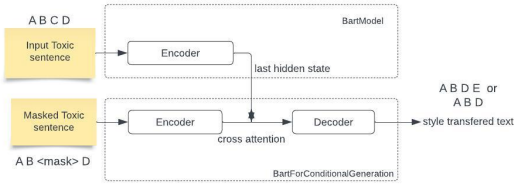

Figure 1: BART Dual Encoders-Decoder Architecture for Text Style Transfer. The architecture includes an encoder from BartModel followed by another encoder-decoder from BartForConditionalGeneration with a language model head (both use BART base parameters). The first encoder takes the original toxic input text which generates a contextualized representation, and a masked sequence is passed to the second encoder for generating style representation. The first encoder's last hidden layer is passed as cross attention to the decoder which is used to generate the output text in the desired neutral style

a translation task, leading us to hypothesize that fine-tuned encoder-decoder models would perform best for text-style transfer. We also constructed a custom BART model (Figure 1) with dual encoders and masked profane words, inspired by BART's pre-training approach of corrupting documents and optimizing a reconstruction loss. We hypothesized that by masking the profane words, we would force the model to remove or replace these words with more appropriate ones during training. This resembles the "word-filtering" and "vocabulary-shifting" techniques discussed in Section 2 and was not included in the original BART single encoder models.

The last approach was a more granular approach, where we built a text-style disentanglement architecture from scratch for text detoxification.

Text-Style Disentanglement We designed a custom architecture using multiple encoders and decoders to separate semantics and style in offensive and neutralized texts. This approach is based on the assumption that sentences can be decomposed into latent representations of semantics and style, similar to Tegene et al. (2023)'s method. With parallel text, we assume shared semantics between toxic and neutralized texts, and that neutralized text maintains the original meaning (Appendix B). During inference, the learned neutralized style is applied to extracted semantics from toxic text, producing neutralized output.

We used simpler architectures for the detoxification task, including independent linear projections of semantic and style representations and an autoencoder with bottleneck latent representations divided between semantics and style. Custom loss

functions were employed for disentanglement, optimizing for similarity loss between semantic vectors, orthogonal loss between toxic style and semantic vectors, and token similarity loss between predicted logits and target text token IDs.

For mixed encoder-decoder architectures like BERT and GPT2, a mapping layer connected encoder tokens to decoder tokens between output and prediction layers.

5 Results and Discussion

5.1 Classification Task for NonToxicScore

After the initial experiment with fine-tuning BERT and DistilBERT (Sanh et al., 2020) model for the classification task, we selected DistilBERT for further improvement due to its comparable performance with BERT (Devlin et al., 2019) while being more efficient. We then explored different model architectures to find the best model for our task. We found that a two-layer feed-forward network on top of the CLS token provided better results than a one-layer network as shown in Table 1. Freezing all of the DistilBERT layers was also more effective, therefore, we proceeded with this two-layer architecture and experimented with sampling methods and token lengths.

Results of the different sampling techniques are shown in Table 2. While undersampling performed slightly better than upsampling with augmentation with 0.2% higher of F1-score on the Jigsaw dataset shown in Table 1, when evaluating on the APPDIA dataset, results in Table 2 show that the scores from undersampling were severely lagging behind upsampling with augmentation by up to -20% when classifying non-toxic text.

For token length, we have found that having shorter tokens generally helps in classifying toxic text on the APPDIA dataset (Table 2). This is because toxicity typically materializes in short sentences and phrases, therefore including more tokens in a sentence generally increases the chances of a "non-toxic" classification as most words are not toxic. Hence, the classification model also performs slightly better with longer tokens for classifying the non-toxic text.

Table 2 demonstrates that the DistilBERT model with data upsampling and augmentation trained on 512 tokens can achieve reasonable performance on both toxic and non-toxic text. Therefore, we chose this model as our final classification model used for extracting the NonToxicScore in style transfer task.

While evaluating the performance of the chosen DistilBERT model in calculating NonToxicScore, we found that the model performed well in identifying sentences that contained toxic words. However, the model could misclassify non-toxic sentences as being toxic, particularly for sentences that sounded authoritative or contained strong opinions in their context which materialized with low NonToxicScore scores as shown in Appendix F. The classification model also had difficulty classifying some toxic texts, particularly for sentences that did not contain offensive words. These sentences achieved high NonToxicScores although their labels were "toxic". Upon closer examination reveals that the real toxicity seems to be grounded in real world context not present in the sentences. For example, the action of "wipes voting machines clean" may not seem toxic by itself, but in the context of the 2020 election, this may provoke people. In another example, the phrase "spewing logical fallacies" may not be toxic by itself since it is just an action, but to understand the connotation of "spewing" as opposed to other verbs such as "accusing" while grounded in who tend to fall for "logical fallacies" in the real world could be considered offensive. These examples highlight the subjectivity of text classification, which can vary among different annotators which may also sway our models.

5.2 Text Style Transfer Task

Comparing the toxic texts and their pair-wise human annotated text from the APPDIA test set, the ROUGE-1 scores was 68.9%, and the NonToxicScore of 79.4% was evaluated on the annotated text (Table 3). The high NonToxicScore was expected since the annotated texts have offensive words removed or completely rewritten by human experts. Due to the bias created from the Jigsaw dataset and discussed in Section 5.1, the NonToxicScore evaluated on the APPDIA dataset was not 100%. It is important to note that this NonToxicScore was highest compared to the outputs of all of our models and serves as the upper-bound benchmark for our models.

When comparing the generated texts from different models with the human annotated texts in Table 3, the encoder-decoder models achieved higher ROUGE scores but settled with lower NonToxicScores than the generative language models. This tradeoff indicates that the encoder-decoder mod-

Models & Configurations	Max Length = 64		Max Length = 512	
	F1 Score	Accuracy	F1 Score	Accuracy
BERT - 1 layer + Undersampling	90.8	93.1	-	-
DistilBERT - 1 layer + Undersampling	92.6	92.6	94.3	94.3
DistilBERT - 2 layers + Undersampling	**93.1**	**93.0**	**94.6**	**94.6**
DistilBERT - 2 layers + Upsampling + Augmentation	**92.9**	**92.9**	**94.4**	**94.4**

Table 1: Toxic and non-toxic classification evaluation on Jigsaw test set with fine-tuned BERT and DistilBERT model.

Model: DistilBERT - 2 layers		APPDIA Toxic Texts		APPDIA Non-Toxic Texts	
Sampling Methodology	Max Length	Accuracy	NonToxic Score	Accuracy	NonToxic Score
Undersampling	64	**93.5**	9.2	66.3	64.5
Undersampling	512	91.0	9.1	63.3	63.0
Upsampling + Augmentation	64	89.0	**7.8**	82.4	78.4
Upsampling + Augmentation	512	88.9	8.4	**83.4**	**79.4**

Table 2: NonToxicScore and classification evaluation on APPDIA test dataset with DistilBERT models. The model trained with 2 hidden layers, upsampling technique and 512 tokens could classify both toxic text and non-toxic text well, and achieve the highest NonToxicScore for the non-toxic text.

els preserved the meaning of the original text well, whereas the generative language models generated non-toxic words better. Fine-tuning the larger models in both architectures yielded better ROUGE scores and NonToxicScores. Fine-tuned BART large achieved the best ROUGE scores with decent NonToxicScore, and fine-tuned GPT-Neo 1.3B achieved the best NonToxicScore with decent ROUGE scores.

Generative Language Models With generative models, we used few-shot GPT2 (Radford et al., 2019) as baseline. Following experiments included zero-shot fine-tuned GPT2, few-shot 2.7B parameter GPT-Neo (Black et al., 2021), and zero-shot fine-tuned 1.3B parameter GPT-Neo.

Without tuning, few-shot GPT2 achieved an impressive NonToxicScore of 57.3% despite having unremarkable ROUGE scores. This was expected, since GPT2 would not know how to detoxify source sentences according to the style of the annotator who manually rewrote those sentences. As seen in the examples in Appendix A, few-shot GPT2 generated sensible and grammatically sound text. However, the sentences that it generated could be very long and sometimes had different meaning from the original input texts. Therefore, the elevated NonToxicScore was a combination of its attempt to detoxify from few-shot examples and of simply having more non-toxic tokens. There were instances that few-shot GPT2 actually made the result more toxic as demonstrated in Table 6 of Appendix A. This could be the result of having more toxic and vulgar language in the few-shot examples selected.

We anticipated zero-shot fine-tuned GPT2 to outperform few-shot base GPT2 in all metrics, which Table 3 confirms. The zero-shot fine-tuned GPT2 generated more neutral text with higher NonToxicScores (Appendix A). Another distinction was that the output length in zero-shot fine-tuned GPT2 closely matched the original text length. While fine-tuning enabled GPT2 to produce less toxic and fluent results closer to the annotated output, the semantics occasionally deviated (Appendix A). For example, "on drugs" in Table 5 was interpreted as substance use instead of "crazy" or "illogical." Thus, while less toxic and closer to the original text, the model missed the correct semantics.

We hoped that the few-shot GPT-Neo model would produce slightly better results than the few-shot GPT2 model, however the results were quite surprising. Given the same prompts as the few-shot GPT2 model, the GPT-Neo model scored quite low in NonToxicScore, failing to remove toxicity. Table 3 shows that the few-shot GPT-Neo achieved the lowest NonToxicScore among all the models attempted. The few examples in Appendix A illustrates the few-shot GPT-Neo model actually exacerbated the toxic tone of the original text by inserting vulgarity into the output with better than expected readability. Our leading hypothesis is that better

Evaluated Annotated Text Against Toxic Text				NonToxicScore on Annotated Text
Benchmark	**ROUGE1**	**ROUGE2**	**ROUGEL**	
Annotated Non-Toxic Text	68.9	56.1	68.3	79.4
Evaluated Generated Text Against Annotated Text				**NonToxicScore on Generated Text**
Models	**ROUGE1**	**ROUGE2**	**ROUGEL**	
Baseline - GPT2 - Few Shot	11.8	1.7	9.9	57.3
GPT2 - Fine tuned	27.8	21.1	27.5	67.4
GPT-Neo 2.7B - Few Shot	29.8	9.6	27.1	37.1
GPT-Neo 1.3B - Fine tuned	61.8	48.2	61.4	**74.9**
T5 - Fine tuned	66.8	54.7	66.4	39.3
BART - Dual Encoders with Masking	59.8	47.9	59.5	57.3
BART base - Fine tuned	69.7	57.8	69.2	57.6
BART large - Fine tuned	**71.0**	**59.3**	**70.5**	58.2

Table 3: Neutralize toxic text model evaluation results on APPDIA test dataset.

prompt-engineering could have improved this approach since it is evident that GPT2 and GPT-Neo contextualized the prompts very differently, such that GPT2 contextualized our prompt more correctly than GPT-Neo. Appendix E is an example of what the few-shot prompt looks like.

Our final generative model experiment involved fine-tuning the 1.3B parameter GPT-Neo model to perform zero-shot inference. We hypothesized that if GPT-Neo could generate such coherent and easy to read text despite the toxicity from few-shot learning, fine-tuning the GPT-Neo model should produce readable, coherent, and non-toxic text. Shown in Table 3, compared to the previous attempts, the results of fine-tuning the 1.3B parameter GPT-Neo model achieved the best NonToxicScore, demonstrating the success in fine-tuning the model. Compared to the other generative models, this approach achieved the highest ROUGE scores in the GPT family. The examples in Appendix A demonstrated that the generative model achieved this by either removing or replacing offensive words and phrases, or negating the antonym of the offensive word. In short, fine-tuning GPT-Neo model resulted in the correct contextualization of the semantics of the original text.

Encoder-Decoder Models For encoder-decoder models, we performed fine-tuning on three main models: T5, BART and the custom dual-encoders BART model.

Fine-tuned T5 model was able to neutralize some of the toxic words by removing them from the sentences or replacing them with synonyms, even though it could not remove all toxic words completely. For example, some negative connotation words still exist in the generated texts such as "stupid", "jerk" and "hate". In absence of toxic words, the model was unable to learn the vocabulary and neutralize the sentence effectively. As a result, the generated texts resembled the original toxic texts closely as demonstrated by the high ROUGE scores and low NonToxicScores in Table 3. This is because most of the generated sentences still have negative connotations or hostile tones. Appendix A provides some examples of the fine-tuned T5 output.

Fine-tuning BART models gave us the best ROUGE scores as shown in Table 3. BART large was able to yield the highest ROUGE scores out of all models, and its generated texts had the most similar tokens to the human annotated texts. However, similar to the T5 model, when toxic words were absent from the original toxic text, the model struggled to neutralize the sentences. The tones for some of the generated sentences are still slightly hostile, negative or sound authoritative. Therefore, the NonToxicScore for BART large is lower than GPT-Neo.

Despite our expectations, the dual-encoder BART model produces ROUGE scores that are slightly lower than the original BART base models, while having a comparable NonToxicScore. The decrease in ROUGE scores can be attributed to the model's tendency to shorten sentences and omit some context from the source text.

5.3 Tradeoffs Generative Language Models vs Encoder-Decoder Model

The choice between generative language models and encoder-decoder models entails a series of tradeoffs. In our research, we found that our best generative language model, the fine-tuned 1.3B parameter GPT-Neo model, excelled in detoxifying text and produced output with a more neutral tone and style. However, this model could generate quite different texts than the original text in some cases, even though the meaning is not significantly different. For example, it translated "Shut up, atheism is gay" to "Please don't talk"; this example demonstrates why the fine-tuned GPT-Neo model yielded lower ROUGE scores than the best encoder-decoder model, the BART large model. On the other hand, the fine-tuned BART large model performed better in preserving semantics as shown by higher ROUGE scores, ensuring that the output remains closer to the original intent. Despite this advantage, the encoder-decoder model was not as effective in detoxifying the text to the same extent as generative models (Figure 2). Consequently, selecting the appropriate model depends on the desired balance between detoxification and semantic preservation, as neither model performs exceptionally well in both domains.

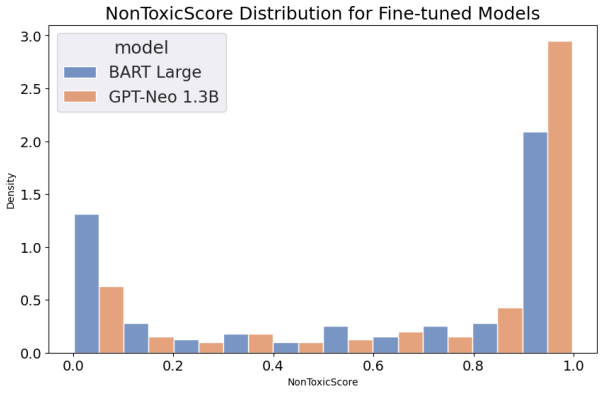

Figure 2: BART Large generates more close-to-zero NonToxicScore texts (not able to detoxify) than the fine-tuned GPT-Neo Model.

6 Next Steps

The text-style disentanglement architecture was an attempt at a novel approach to this problem and more work needs to be done to successfully disentangle semantics from tone. We have learned from our experiments that the disentanglement architecture would have to retrain either the decoder or the encoder from scratch, since we proposed to use BERT encoder and GPT decoder. Two alternatives to this approach is to pretrain a heavy mapping layer between BERT outputs to GPT tokens or leverage the BART tokenizer.

In addition, since this is a novel architecture, the decoder layers would need to be trained on large corpus in order to sound grammatically correct and readable. That would require pre-training on large corpus' such as Wikipikea.

7 Conclusion

In the contemporary online landscape, text detoxification has emerged as a critical issue, as improper content moderation faces scrutiny and potential censorship accusations. We addressed this problem by considering it as a general text-generation issue using generative models and as a translation problem employing encoder-decoder architectures like T5 and BART. Our experiments revealed that decoder-only generative models produced more fluent, albeit semantically altered, results, while encoder-decoder models maintained semantic correctness at the expense of detoxification quality. Although the optimal approaches identified in this research involve tradeoffs, we aspire to eventually discover a solution that eliminates the need to compromise between semantic preservation and detoxification.

References

Diana L. Ascher and Safiya Umoja Noble. 2019. Unmasking Hate on Twitter: Disrupting Anonymity by Tracking Trolls. In *Free Speech in the Digital Age*. Oxford University Press.

Katherine Atwell, Sabit Hassan, and Malihe Alikhani. 2022. APPDIA: A discourse-aware transformer-based style transfer model for offensive social media conversations. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 6063–6074, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

Alex Beutel, Jilin Chen, Zhe Zhao, and Ed H. Chi. 2017. Data decisions and theoretical implications when adversarially learning fair representations.

Sid Black, Gao Leo, Phil Wang, Connor Leahy, and Stella Biderman. 2021. GPT-Neo: Large Scale Autoregressive Language Modeling with Mesh-Tensorflow. If you use this software, please cite it using these metadata.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners.

Yu Cheng, Zhe Gan, Yizhe Zhang, Oussama Elachqar, Dianqi Li, and Jingjing Liu. 2020. Contextual text style transfer. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2915–2924, Online. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding.

Samuel Gehman, Suchin Gururangan, Maarten Sap, Yejin Aguilar, and Myle Ott Bernstein. 2020. Realtoxicityprompts: Evaluating neural toxic degeneration in language models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 87–110. Association for Computational Linguistics.

Shagun Jhaver, Darren Scott Appling, Eric Gilbert, and Amy Bruckman. 2019. "did you suspect the post would be removed?": Understanding user reactions to content removals on reddit. *Proc. ACM Hum.-Comput. Interact.*, 3(CSCW).

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. 2019. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Edward Ma. 2019. Nlp augmentation library (nlpaug). Available at `https://github.com/makcedward/nlpaug`.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI Blog*.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2018. Language models are unsupervised multitask learners.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2020. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter.

Abebe Tegene, Qiao Liu, Yanglei Gan, Tingting Dai, Habte Leka, and Melak Ayenew. 2023. Deep learning and embedding based latent factor model for collaborative recommender systems. *Applied Sciences*, 13(2).

Jason Wei and Kai Zou. 2019. EDA: Easy data augmentation techniques for boosting performance on text classification tasks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 6382–6388, Hong Kong, China. Association for Computational Linguistics.

Appendix

A Appendix A - Sample outputs

More examples showing the pros and cons of different models

Models	Output sentences	Non Toxic Score
Original Text	People publicly shame smokers for their bad choices. Why are fat people exempt from criticism?	19.8
Human Annotated Text	People publicly shame smokers for their bad choices. Why are overweight people exempt from criticism?	73.4
Baseline - GPT2 - Few Shot	I don't care if you're fat or not. I just want you to know that I'm not going to let you down.	19.7
GPT2 - Fine tuned	ers are exempt from criticism. Why are fat people exempt from criticism?	67.4
GPT-Neo 2.7B - Few Shot	People publicly shame people who smoke for bad decisions. Why is fat a protected class?	11.5
GPT-Neo 1.3B - Fine tuned	People publicly shame smokers for their bad choices. Why are people exempt from criticism?	73.5
T5 - Fine tuned	People publicly shame smokers for their bad choices. Why are fat people exempt from criticism?	19.8
BART - Dual Encoders	People publicly shame smokers for their bad choices. Why are these people exempt from criticism?	75.8
BART base - Fine tuned	People publicly shame smokers for their bad choices. Why are fat people exempt from criticism?	19.8
BART large - Fine tuned	People publicly shame smokers for their bad choices. Why are fat people exempt from criticism?	19.8

Table 4: Example of models' outputs when the original text does not contain offensive words

Models	Output sentences	NonToxicScore
Original Comment	the f***.... you are on drugs	0.1
Annotated Text	uh..... you are not being realistic	98.3
Baseline - GPT2 - Few Shot	I don't care what you think about me, I'm here to help you out. I am here for you.	99.3
GPT2 - Fine tuned	ive been on drugs for years	96.2
GPT-Neo 2.7B - Few Shot	the f***ing f*** are you on?	0.1
GPT-Neo 1.3B - Fine tuned	What.... are you on drugs	56.4
T5 - Fine tuned	you are on drugs	78.1
BART - Dual Encoders with Masking	the U.S. you are on drugs	66.6
BART base - Fine tuned	you are on drugs	78.1
BART large - Fine tuned	the f***.... you are on drugs	0.1

Table 5: Example of models' outputs when the original text contains offensive words and the semantics of the original text could be interpreted with different meanings.

Models	Output sentences	NonToxicScore
Original Comment	So maybe you should be more retarded.	1.5
Annotated Text	So maybe you should be more backward	99.0
Baseline - GPT2 - Few Shot	I don't know what to do with you. You're just a f***ing moron. I'm just going to give you the benefit of the doubt.	0.1
GPT2 - Fine tuned	ive never seen anything like this	94.7
GPT-Neo 2.7B - Few Shot	So you're a f***ing retard.	0.1
GPT-Neo 1.3B - Fine tuned	So maybe you shouldn't be smart	99.4
T5 - Fine tuned	So maybe you should be more retarded.	1.5
BART - Dual Encoders with Masking	So maybe you should be more careful.	99.7
BART base - Fine tuned	So maybe you should be more mature.	99.0
BART large - Fine tuned	So maybe you should be more backward.	99.0

Table 6: Example of models' outputs when the toxic word in the original text may or may not be interpreted as offensive by itself.

B Appendix B - Dataset

Below are a few examples from the training data set. We've redacted some words with *'s for viewer discretion. Note the similarity of the semantics between the offensive-text and the style-transferred-text as well as the complexities inside the text itself, such as emojis and incorrect grammars.

offensive-text	style-transferred-text
Ok, this makes no sense. This will create vigilantes. And I hope they put this savage animal down before he hurts anyone else.	Ok, this makes no sense. This will create vigilantes. And I hope they take care of this uncontrollable person before he hurts anyone else
This is really disgusting. People not able to compete even in the 3rd men league getting paid twice as much.	This is really off-putting. People not able to compete even in the 3rd men league getting paid twice as much.
Good lord you people are f***ing stupid.	Good lord you people don't use your brain
Yeah, no. And for that f*** you.	Yeah, no. And for that go away.
Holy sh** this sub is f***in gross	Holy moly this sub is gross
Oh joy can't wait for you dumb f***s to misunderstand this science too. 🙄	Oh joy can't wait for all of you irrational people to misunderstand this science too. 🙄

Table 7: Sample data from APPDIA dataset

C Appendix C - NonToxicScore

Example of how NonToxicScore was calculated. The first three examples are from the toxic texts, and the last three are from the human annotated non-toxic texts.

Example Text	NonToxic Score	Predicted Label	Explanation
Ok, this makes no sense. This will create vigilantes. And I hope they put this savage animal down before he hurts anyone else.	10.2	Toxic	Negative meaning
This is really disgusting. People not able to compete even in the 3rd men league getting paid twice as much.	4.1	Toxic	Negative meaning &bad word ("disgusting")
Good lord you people are f***ing stupid.	0.1	Toxic	Bad word (f***ing stupid)
Yeah, no. And for that go away.	98.5	Non-Toxic	No bad word
Holy moly this sub is gross	18.1	Toxic	Bad word ("gross")
Oh joy can't wait for all of you irrational people to misunderstand this science too. 🙄	51.3	Non-Toxic	Negative meaning
Average NonToxicScore	30.4		

Table 8: Sample data from APPDIA dataset

D Appendix D - Text Data Augmentation

Example of data augmentation with nlpaug used in upsampling technique. Note that data augmentation can sometimes lead to slight changes in tone, as shown in the example above. This is a necessary tradeoff we made with upsampling text data so that upsampled data are not duplicates of the original data.

Original data	Zip It! I don't care to hear your whining. Clearly, you are here to cause discord, not to contribute anything useful to the project.
Augmented data	zip it! i don't care to hear your question. clearly, you get here to cause discord, lets just contribute anything useful to the project.

Table 9: Sample of augmented data used in upsampling for training classification models

E Appendix E - Prompt for Few Shot Learning

Sample prompt for few shot learning:

> Rewrite the toxic text in non-toxic style:
> ###
> Toxic text: So maybe you should be more retarded.
> Non-toxic text: So maybe you should be more backward
> ###
> Toxic text: You don't deserve it dumb*ss.
> Non-toxic text: You don't deserve it.
> ###
> Toxic text: Youre soft as baby sh**
> Non-toxic text:

F Appendix F - Missed Classifications on APPDIA Dataset

Example of wrong classifications using the best DistilBERT model on APPDIA test dataset

Texts	Label	Predicted	NonToxicScore
There's plenty of evidence, it's exactly why they won't allow forensic audits *wipes voting machines clean of their records*	Toxic	Non-Toxic	97.6
Also there are women like this but they don't write about it like this on ducking reddit.	Toxic	Non-Toxic	99.0
Do you have a substantial counter argument or are you just spewing logical fallacies?	Toxic	Non-Toxic	87.3
Stop dude. Worst of the worst!	Non-Toxic	Toxic	33.3
They literally steal and sell babies in convenience stores. Sorry but most are undesirables. (a small % escape their culture and integrate)	Non-Toxic	Toxic	28.6
I would not say nuts...it just looks very suspicious. Either way, the dude is totally unacceptable.	Non-Toxic	Toxic	19.9

Table 10: Example of wrong classifications using the best DistilBERT model on APPDIA test dataset. The model mostly classifies a comment as toxic when seeing negative words and meaning in the context.