RoboCop:
Remediation Of Biased Observations and Commentary for Optimal Politness
A Reinforcement Learning Approach to Detoxifying Hate Speech

Luca Carroz [1] **Baptiste Maquignaz** [1] **Xavier Ogay** [1]

Abstract

Although some models have demonstrated excellent performance in detoxifying tasks, previous research has established that maintaining the original meaning of toxic sentences is often challenging. In this project, we explored the use of reinforcement learning to train a BART model [1] to preserve strong semantic and contextual meaning while detoxifying sentences. We observed that since toxic and detoxified translations often differ by only a few words, naively training the model on parallel data, as done in [2], quickly led to the model simply copying the input sentence, resulting in small loss. To address this issue and prevent the model from converging on this trivial behavior, Paradetox required training for 10000 epochs. Due to limited resources and time, we incorporated a toxicity classifier to penalize the reinforcement learning reward when the model generated harmful or vulgar content. This approach aimed to expedite training and avoid the naive copying behavior that can still be observed in some cases during inference.

Keywords: Detoxifying, Reinforcement Learning, NLP, meaning preservation.

1. Introduction

Handling online hate speech has become a significant challenge in recent years and AI-based moderation tools can be valuable assets in promoting a safer and non-offensive online space. Researchers have already provided detoxifier models such as Paradetox [2] or GreenLlama [3] which provide high-quality detoxification of offensive or inappropriate sentences. However, both of these models face a notable issue: often, the models prioritize detoxification at the expense of preserving the original meaning. This means that during inference, the model will sometimes provide a general non toxic statement that is far from the toxic sentence. The objective of our work is then to obtain an insight on whether the use of reinforcement learning could help a producing meaning-preserved detoxified outcomes. Additionally to that we also used a toxicity classifier [4] to penalize the toxic outcomes in an attempt to avoid simple repetitions of the input sentence.

Measuring meaning preservation

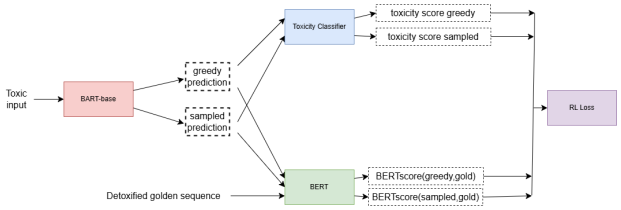
Measuring meaning preservation has always been a concern in Natural Language Processing because language is by construction extremely complex. Many metrics have been developed in order to capture how 'close' two sentence are. Among the most used ones, we can find BLEU [5], ROUGE [6], BertScore [7], METEOR [8] and many more. In our case, we judged that the most important would be to measure the semantically closeness and deep contextual meaning of the sentence, we then naturally thought of using BertScore as our metric to measure meaning preservation. BertScore compares the cosine similarity of the contextual BERT embedding of two sequences and therefore is often used when capturing deep semantic links is required.

Reinforcement Learning

Now that we have a metric to measure meaning preservation, we can use it to encourage our model to keep the sentence meaning by providing BERTScore as a reward. This is when reinforcement learning comes in place: during training, additionally to the usual greedy model prediction, we will generate an alternative stochastic sequence following the probability distribution of our model. This random sequence will then be compared to the greedily generated one via the BERTScore metric and the toxicity classifier. If the stochastic sequence have a better BERTScore and is less toxic, we encourage this behavior. If on the other hand the stochastic sequence achieves lower BERTScore or higher toxicity, we will want to keep the original behavior.

[1]Group 3.

EE-559: *Deep Learning*, 2024.



2. Related Work

The first works that tried tackling online detoxification considered it as a variant of style transfer task. Publications such as [9] explored the problem in an unsupervised manner. The absence of parallel dataset has been a barrier to further improve the detoxification efficiency as no supervised learning could be performed without a high-quality dataset.

In 2022, Paradetox [2] was published. $10'000$ non-toxic paraphrases were collected by human annotators, constituting a parallel dataset that would facilitate supervised training. Although the main purpose of the paper was to create the dataset, the authors still fine-tuned a BART model to provide a concrete example of use of their dataset. The authors trained the LLM for $10'000$ epochs, achieving state of the art results.

In 2024, [3] built a model that would beat the Paradetox BART at detoxification. The authors proposed a ChatGPT 3.5-turbo based pipeline to transform nonparallel datasets initially meant for toxicity detection into pairs that could be exploited in a similar manner as the Paradetox dataset. This huge quantity of data would then allow them to fine-tune a much bigger LLaMA-2 model. Furthermore, to encourage Chain Of Thought reasoning, they used ChatGPT again to generate an explanation of the toxicity reasons that would be additionally provided as context to their models. GreenLLaMA also offers a paraphrase detector to filter out the non-detoxifiable cases, allowing it to abstain instead of writing harmful content while hallucinating. Although the GreenLLaMA model performs incredibly well, the author highlight a common limitation to the existing detoxifier models, including theirs. Detoxifiers seem to have trouble preserving meaning from the original sequence especially with adversarial examples. This problem was the starting point of our project.

Knowing that we did not have access to a lot of compute power or time and that we had only few experience in natural language processing, we chose to build our experiment on the fine-tuning proposed in the Paradetox paper. However, since the authors trained for so many epochs, it could take weeks to reproduce with our limited resources. Therefore, we also tried speeding up the training process by improving the objective loss.

3. Method

We encountered several limitations, a significant one being the lack of computational resources and time, which constrained our ability to work with larger models like those used in projects such as GreenLlama, which utilize a GPT framework. Given these constraints, we focused on smaller, more manageable models. Ideally, we would have employed a teacher-student approach using a GPT model version 3.5 or higher, renowned for their robust outputs, but this was not feasible due to the necessity of paid API access, which we opted not to pursue.

Consequently, we adjusted our expectations and selected a more budget-friendly, fine-tunable model such as the BART-base[1], one of the smallest BART model available (i.e. 140M parameters), aligning closely with the methodologies described in the Paradetox paper. Additionally, the use of the *peft* library helped to reduce our training load.

Another limitation arose from observations made in the GreenLlama findings, which indicated that instruction-tuned large language models (LLMs) often fail to adhere to directives when handling toxic inputs, defaulting to generic responses. To address this, we concentrated on implementing a reinforcing loss mechanism aimed at producing polite outputs while preserving the core meaning of the original sentences, enhanced by our custom Bert/Politeness reinforcement learning loss defined as product of the difference between the rewards and baseline rewards, and the sum of the negative log probabilities of predicted sequence tokens y_t^s given the previous tokens and the input x:

$$RL_{Loss} = -\left[\lambda_{tox}(s^s_{tox} - s^g_{tox}) + \lambda_{bert}(s^s_{bert} - s^g_{bert})\right] \cdot \sum_{t=1}^{n} \log p(y_t^s | y_1^s, \ldots, y_{t-1}^s, x) \quad (1)$$

Where: s^s_{tox} and s^g_{tox} are the politeness scores of the sampled and greedy sequences, respectively.

s^s_{bert} and s^g_{bert} are the BERTScores of the sampled and greedy sequences, respectively.

λ_{tox} and λ_{bert} are the weights assigned to the politeness score and the BERTScore, respectively.

The total loss that is optimized during training is a weighted sum of the MLE and RL losses:

$$\text{Total Loss} = \text{MLE Loss} + \alpha \cdot \text{RL Loss}$$

where α is the weight given to the RL loss.

The politeness score is calculated by the R4 Target model[4]

4. Validation

To be able to compare our results with the ones from the authors of Paradetox [2], we first trained a model on 100

epochs without the reinforcement loss. The resulting model showed relatively good style accuracy and similarity, but bad fluency and BLEU score. With such a small number of epochs, it seems that the model has enough time to locate the toxicity in a sentence and get rid of it without changing the meaning of the sentence too much, but it struggles in writing a grammatically correct sentence, hence the low fluency score.

We went further and trained the model for 1000 epochs, hoping that the model would now have enough time to learn to construct fluent sentences, but the model actually starts to over-fit after only 100 epochs. Since the Bert model from Paradetox is trained for 10000 epochs without over-fitting, we hoped that we would not over-fit so fast. Our best explanation to this early over-fitting comes from the fact that some of our hyperparameters might be sub-optimal. Indeed, we could not fine-tune all our hyperparameters since we have 5 of them and a single training take approximately 12 hours on a T4. Maybe a more suited learning rate or scheduler might have delayed the over-fitting, but from the ones we tried the current setting was the best.

Our third model was first trained without the reinforcement loss for 100 epochs, and then for an additional 40 epochs with the rl loss. The considerable overhead in computation induced by the rl loss slowed down the training by a factor of approximately 50, which explains why we only trained for 40 epochs with the rl loss. The effect of the rl loss was almost insignificant even after 40 epochs, which might be due to a bad choice of hyperparameters but most likely to the fact that Bart cannot produce its output tokens one by one, and we thus had to sample each token independently from the others, which in general leads to sentences with less meaning. An idea that we did not have time to pursue would be to use a different model that can produce its tokens one by one in order to produce sentences with more meaning.

Our results are displayed in Table 1. The metrics for $Paradetox$ are directly taken from their paper, $Ours_1$ and $Ours_2$ are our two models trained on 100 and 1,000 epochs respectively (without rl loss), and $Ours_3$ is trained on 100 epochs without rl loss and then 40 epochs with rl loss. The metrics we compute are the same as in [2] namely:

- *BLEU*: The BLEU score [5]
- *Style accuracy* (STA): The percentage of nontoxic outputs identified by a style classifier. We take the same classifier as in [2] for consistency.
- *Content preservation* (SIM): The cosine similarity between the embedding of the original text and the output computed in the embedding space of a model trained to associate paraphrases.
- *Fluency* (FL): The percentage of fluent sentences identified by a RoBERTa-based classifier. We also take the same classifier as in [2] for consistency.

MODEL	BLEU	STA	SIM	FL	J
PARADETOX	64.53	0.89	0.86	0.89	0.68
$Ours_1$	47.56	0.71	0.75	0.35	0.19
$Ours_2$	43.32	0.87	0.68	0.17	0.10
$Ours_3$	46.78	0.73	0.76	0.33	0.18

Table 1. EVALUATION OF DETOXIFICATION MODELS

The joint metric J is the multiplication of the 3 last metrics. We see that our modest models still manage to almost match the best model from Paradetox in terms of style accuracy and content preservation, but it is much worse in terms of fluency. Note that in their paper, the authors of Paradetox manually inspected their test set to remove any sentence that was ill-conditioned (by example if the whole sentence is toxic, like a single swear word). Obviously, this is a very tedious task that is not intellectually interesting and whose only goal is to have better metrics. We chose not to do that manual inspection and hence, our metrics are probably a bit worse than what would be if we filtered our test set.

5. Conclusion

This study has explored the potential of reinforcement learning to address the challenge of preserving semantic integrity while detoxifying online content. Our approach, which integrates a BART model with a toxicity classifier and BERTScore-based reinforcement, aimed to maintain the balance between content fidelity and non-offensiveness. Despite the constraints posed by limited computational resources and the use of smaller model architectures, our findings show the feasibility of achieving notable detoxification outcomes with enhanced meaning preservation.

The experimental results, achieved under resource-limited conditions, suggest that reinforcement learning can be a valuable strategy in the development of language models tailored for specific tasks such as content moderation. Although our models did not excel in fluency as compared to more extensive projects like Paradetox, they performed commendably in style accuracy and content preservation metrics.

We acknowledge the limitations of our approach, primarily related to the size of the model and the duration of training. Future research could focus on expanding the model's capabilities by employing more robust architectures and longer training periods, but also on trying to speed up the computation of the reinforcement learning loss, which is the bottleneck for the moment to really assess how usefully the rl loss is.

References

[1] M. Lewis, Y. Liu, N. Goyal, M. Ghazvininejad, A. Mohamed, O. Levy, V. Stoyanov, and L. Zettlemoyer, "Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension," *arXiv preprint arXiv:1910.13461*, 2019.

[2] V. Logacheva, D. Dementieva, S. Ustyantsev, D. Moskovskiy, D. Dale, I. Krotova, N. Semenov, and A. Panchenko, "Paradetox: Detoxification with parallel data," in *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 6804–6818, 2022.

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

[4] B. Vidgen, T. Thrush, Z. Waseem, and D. Kiela, "Learning from the worst: Dynamically generated datasets to improve online hate detection," *arXiv preprint arXiv:2012.15761*, 2020.

[5] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "Bleu: a method for automatic evaluation of machine translation," in *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics* (P. Isabelle, E. Charniak, and D. Lin, eds.), (Philadelphia, Pennsylvania, USA), pp. 311–318, Association for Computational Linguistics, July 2002.

[6] C.-Y. Lin, "ROUGE: A package for automatic evaluation of summaries," in *Text Summarization Branches Out*, (Barcelona, Spain), pp. 74–81, Association for Computational Linguistics, July 2004.

[7] T. Zhang, V. Kishore, F. Wu, K. Q. Weinberger, and Y. Artzi, "Bertscore: Evaluating text generation with bert," *arXiv preprint arXiv:1904.09675*, 2019.

[8] S. Banerjee and A. Lavie, "METEOR: An automatic metric for MT evaluation with improved correlation with human judgments," in *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization* (J. Goldstein, A. Lavie, C.-Y. Lin, and C. Voss, eds.), (Ann Arbor, Michigan), pp. 65–72, Association for Computational Linguistics, June 2005.

[9] C. N. d. Santos, I. Melnyk, and I. Padhi, "Fighting offensive language on social media with unsupervised text style transfer," *arXiv preprint arXiv:1805.07685*.