RAG Meets Detox: Enhancing Text Detoxification Using Open Large Language Models with Retrieval Augmented Generation

Notebook for PAN at CLEF 2024

Erik Řehulka[1], Marek Šuppa[1,2]

[1]Comenius University, Bratislava, Slovakia
[2]Cisco Systems

Abstract

In this work we present our solution at the Multilingual Text Detoxification 2024 task, whose objective is to take toxic text and convert into one that conveys the same meaning without containing any toxicity. Our approach utilizes open Large Language Models extended with dynamic prompt creation combined with Retrieval Augmented Generation. The evaluation results show that despite its simplicity, our method has the potential to provide competitive results, as evidenced by both the automatic and manual evaluation executed by the task organizers. Overall, our approach ranked 5th in the manual evaluation, with our best-performing language, German, even surpassing the human reference.

Keywords

PAN 2024, Retrieval Augmented Generation, text detoxification, Llama3, LLM, toxic text

1. Introduction

The task of identification of toxicity in texts is an active area of research. Social networks are trying to address this problem by simply blocking such texts. A more interesting and effective approach might be to automatically rewrite these texts, so that they are ideally no longer toxic, but their meaning is kept intact. This processed is denoted as *detoxification*.

The Multilingual Text Detoxification (TextDetox) 2024 task aims to create and explore such methods. The participants are provided with a dataset of toxic texts in several languages from all over the globe, which then should be detoxified. The goal is to find a method, which after evaluation provides texts which are neutral, but their meaning is the same as the toxic text on the input.

We explore how a data scientist with only an API access to a Large Language Model (LLM), in our case Llama3 can develop effective solutions for this task. We did not fine-tune or alter in any way, the only approach was to creatively adjust prompts given to the LLM, so that its outputs will get highest score possible. We have developed several methods, from the simple ones like zero shot prompting, to utilizing existing datasets of text detoxifications and generating these prompts dynamically considering the input text to be detoxified.

For this we have used external tools like vector databases containing pairs of toxic texts and their neutral counterparts, which were queried using embedding of the toxic texts. We have found this method to be competitive and despite its simplicity it achieved high ranking in this task.

We submitted our results under the usernames *erehulka* and *mareksuppa* to the CodaLab portal. Our best-performing languages, in comparison to other participants' submissions, were German and Chinese. Notably, our score for German even surpassed the human references in the manual evaluation. Overall, our approach ranked 5th in the manual evaluation. A more detailed discussion of our results is provided in Section 6.

CLEF 2024: Conference and Labs of the Evaluation Forum, September 09–12, 2024, Grenoble, France
rehulka3@uniba.sk (E. Řehulka); marek@suppa.sk (M. Šuppa)

2. Related Work

Most of the work in this area has been done mainly for the English and Russian language. The first study on automatic detoxification of Russian language was introduced in 2021 [1], where two models were suggested: a BERT-based approach for local corrections and a supervised approach based on a pretrained GPT-2 model. Another contribution presented two unsupervised methods for text detoxification [2]. The first method combined style-conditional language models with paraphrasing models to perform style transfer, while the second method used BERT to replace toxic words with non-offensive synonyms.

In the realm of parallel data collection for the task of detoxification, a ParaDetox pipeline was introduced [3]. It collected non-toxic paraphrases of toxic sentences. This work provided new parallel corpora for training detoxification models, demonstrating that models trained on this data outperformed existing unsupervised approaches by a significant margin.

The challenges of multilingual and cross-lingual detoxification were explored by investigating the behavior of large multilingual models [4]. They found these models can do multilingual style transfer, but it is not possible to do cross-lingual detoxification without fine-tuning.

The RUSSE-2022 shared task [5] was a competition similar to this one, focused on detoxification methods for Russian, featuring parallel training data and manual evaluation. The study found that the best performance for the Russian language was achieved using the ruT5 model, which was fine-tuned on parallel data. In this context, parallel data refers to each toxic record having a corresponding neutral counterpart.

A study has been made in the area of evaluating the quality of the detoxification process, suggesting ChrF and BertScore metrics can be used for this task [6]. The ChrF measure is used even in this task for the computation of the similarity of meaning between the original toxic text and its detoxified version.

3. Datasets

In our work we utilized two datasets, both published by the competition organizers. Both of these datasets are available on the HuggingFace platform.

The first dataset is `textdetox/multilingual_toxic_lexicon` [7], which contains a set of toxic words for each of the input languages. The number of toxic words for each language differed, with the exact numbers depicted in Table 1. As Table 1 shows, these numbers differ a lot between the languages. For example for languages like Amharic, Hindi or even German there are only a couple hundreds of the records, and for the Russian language there is around 141,000 records. These words were used as a lexicon of toxic words, which were passed to some of the prompts, as examples of words which must not be present in the resulting detoxified text.

Table 1
Number of Records per Language

Language	Abbreviation	Number of Records
Amharic	am	245
Spanish	es	1,200
Russian	ru	141,000
Ukrainian	uk	7,360
English	en	3,390
Chinese	zh	3,840
Arabic	ar	430
Hindi	hi	133
German	de	247

The second dataset is `textdetox/multilingual_paradetox` [8], which was published by the organizers after the development phase has ended and contains both the original as well as detoxified inputs for the development phase. We have used this dataset to dynamically create the prompt using

Retrieval Augmented Generation (RAG), more of which is discussed in Section 4.4. For each language the dataset contained exactly 400 records.

4. System Description

On input our system receives a toxic text to be detoxified, along with an abbreviation of the language of the input. All of the languages present on the input with their respective abbreviations can also be seen in Table 1.

Our approach was to create a prompt which then would be passed to the Llama3 model, such that its result would be the detoxified text. Here we present the different methods we used, starting from a simple zero-shot prompting, and finally using retrieval augmented generation with a simple lexicon.

4.1. Zero Shot

In the first phase we explored how efficient is zero-shot approach with just the instructions on how to detoxify the input, with the input at the end. The prompt can be seen in Appendix A and only specifies the task of detoxification. When it comes to different languages, there was no specification on what is the language, just that the result must be in the same language as the input.

With just this knowledge the model outputs were sometimes very different from the input, because it tried to rewrite the whole sentence, to keep the meaning and not be toxic. However the specification of toxic text is very broad, so we needed to somehow specify how the output should look for given input, so it will just rewrite the toxic parts and ideally keep the rest of the text as it is. Thus we have moved to providing examples of input-output pairs for the model.

4.2. Few Shot with language specification and examples

Next we added some examples mainly in English, and also in other languages to the prompt, along with specifying what is the language of the input. The prompt can be seen in Appendix B, with the `{language}` filled in from Table 1. This has proved to obtain better results in languages other than English.

The English examples during the development phase were actually retrieved from the `s-nlp/paradetox` dataset on the HuggingFace platform, which was published before the competition and mentioned in the competition guidelines. Because there was such dataset only for English and Russian (`s-nlp/ru_paradetox`), the examples for other languages were actually used from the output of our model.

4.3. Separate prompts for each language

We have also experimented with creating a separate prompt for each language, with the examples only from the language. We tried to even translate the whole prompt from English to the given language. However this has not proved to be a great improvement, for some languages as Hindi or Amharic it was even worse than the previous approaches, so we have dropped this idea, and for the most languages we have created the prompt in English, with only the examples of input and output being in the target language.

Since these methods utilized no outer knowledge not known to the model, we needed to somehow fill in the examples from real data, which contained pairs of toxic text, and its detoxified version for each language. This dataset has been released after the development phase, containing 400 such examples for each language. With this we have moved to utilizing Retrieval-Augmented Generation and generating the examples such that they will be most similar to the text we are actually trying to detoxify.

4.4. Retrieval Augmented Generation (RAG)

RAG, or Retrieval-Augmented Generation, enhances the capabilities of large language models (LLMs) by incorporating references from knowledge bases external to their training data. These LLMs are

trained on extensive datasets and utilize billions of parameters to perform tasks such as answering queries, language translation, and text completion. The main difference from model fine-tuning is that fine-tuning involves adjusting the parameters of a pretrained language model to adapt it to a specific task or domain. This process typically involves training the model on a smaller, task-specific dataset to refine its understanding and performance in that particular area. Fine-tuning essentially tweaks the existing parameters of the model to specialize its capabilities.

On the other hand, RAG incorporates references from external knowledge bases to enhance the output of a language model without retraining it. Instead of modifying the model's parameters, RAG augments its generation process by retrieving relevant information from authoritative sources and fills the prompt by adding this retrieved information. This approach is not only cost-effective but also ensures that LLM-generated content maintains relevance, accuracy, and utility across diverse contexts.

For this reason, we have chosen to explore the performance of RAG in this specific task, so we won't have to fine-tune existing models, but use them in their current state.

4.4.1. Prompt Template

The complete prompt is available in Appendix C. It consists of several elements, most of which are dynamically generated according to the specific input. However, the prefix, which directs the LLM to act as a *text detoxifier* remains consistent across all prompts:

```
## Task

You are a text detoxifier. On input you receive a text which may be toxic or
    harmful. Your task is to rewrite this text in a way that does not
   contain any toxicity or harmful words, while preserving the original
   content and context.

The Output contains only the detoxified text and nothing else like notes or
   additional information. You do not add any more context to the resulting
    text, which is not in the original text.
Do not rewrite the original text too much, just either remove the toxic part
    completely, or replace it with some non-toxic words while preserving
   the meaning and context.

The language of the input is {language} and the language of the response
   must be the same.
```

The language (in the prompt as `{language}`) is filled in based on the language of the input, as mentioned before.

The most of the remaining prompt is generated based on the specific input to be detoxified.

4.4.2. Retrieval augmentation

In the used prompt, there is also a space for some examples, mainly in the following part:

```
## Toxic words

The input text may contain offensive or harmful words. You should either
   remove them or replace them with non-offensive words.

Here are some examples of toxic words you may find in the input text:

{toxic_words}
```

```
These CAN NOT be used in the output text. You must replace them with non-
    toxic words. If that is not possible, remove them completely.

## Examples

{examples}
```

As the listing shows, the prompt contains parts denoted as `## Toxic words` and `## Examples`, which contain a number of examples that are filled in based on the input text. For this we have used the method of Retrieval Augmentation, where we get a number of examples for both parts relevant for the input text.

For the `## Toxic words` part we use the `multilingual_toxic_lexicon` dataset, from which we get all of the words from the input text present in the lexicon for the input language, along with five random words from this dataset, to make sure that there will always be at least five examples of toxic words. For the selection of words present in the input we have simply filtered those words, which are exactly present in the input text, without creating an index or otherwise. Each of this word is added to the prompt in format `- {word}\n`.

For the second part with `## Examples` we use RAG. We utilize the Chroma vector database [1] to create an index of embeddings from the mentioned `textdetox/multilingual_paradetox` dataset. We generate the index using the LaBSE transformer model, since this model has been used by the organizers in evaluating the submissions. Each record from the dataset is saved in the collection with following fields:

- embeddings - embedded toxic sentence,
- documents - unmodified toxic sentence,
- metadatas - neutral (detoxified) sentence for that row, along with the language of the input,
- ids - necessary id for the database.

Instead of creating a separate index for each of the languages, we have created only one and then retrieve data from the index by specifying which language we want to retrieve. At inference time the same embedding model on which the model was created provides embedding of the sample being evaluated and this embedding is used to query the database, which will return `k` closest items from its index for the same language as the sample. We have set the value of `k` to 10. After getting these closest items, we add them to the prompt, in format:

`Input: {text}\nOutput: {metadata['neutral_sentence']}\n\n`,

with `text` being the original toxic sentence and `neutral_sentence` being detoxified.

4.4.3. Delete baseline

For the Amharic language, this approach has not proved to be ideal. The responses of the Llama3 model for the prompts took much longer than for the other languages, and the resulting score was much lower compared to other participant's submissions. Because of this, we have used for this specific language an approach similar to the one used in delete baseline, that is just deleting toxic words from the input and keeping the rest as is.

For this we have removed all words from the input, which were present in the toxic words lexicon dataset for the Amharic language, and returned this modification as the detoxified text. Nothing else has been done on this particular language, thus it has not been passed through the same pipeline as the one mentioned in this section.

[1] https://www.trychroma.com/

5. Output cleanup

Since the model not always outputs only the detoxified texts, we needed to cleanup the result, so that only the detoxified text would be present.

The model sometimes included a “Note“ saying something about the detoxified text, or it would prefix the result with “Translation:“ or “Output“. We have removed these texts completely, and also for the notes we even removed all text which was after the “Note“ keyword. This is because the note was always at the end of the model output, so we have not lost any information this way. This has been present mainly in the first approaches like zero-shot, for the last part with RAG we have not used this.

Also the model sometimes responded with the text in quotes, so in that case we have removed the quotes from beginning and end of the response. For several languages the format of the quotes was different, so we remove these types of quotes: `"`, `“`, `«`, `’`.

6. Results and Discussion

In this section we describe the main results obtained as part of the Task, in which the outputs of the submitted systems were evaluated both automatically as well as by manually labeling a subsample of 100 texts per language via crowdsourcing.

To evaluate the results, three criteria are assessed:

1. The absence of offensive content or toxic language in the text. For this purpose, a specifically fine-tuned *xlm-roberta-large* model for toxicity binary classification task is used [9].
2. The preservation of the original meaning in the detoxified text. This is quantified by calculating the cosine similarity between LaBSE embeddings.
3. The grammatical correctness of the detoxified text. This is evaluated using the *ChrF* metric [10].

Each of these metrics ranges from 0 to 1. To derive a composite metric, the final score for a given result is computed as the product of all three metrics, referred to as the *joint* metric. Subsequently, the scores for individual languages are calculated as the mean of all scores for that language, and the overall score is obtained by averaging the scores across all languages.

The results of the automatic evaluation can be seen in Table 2. In it, we outline all of the system configurations in order to show their impact on the final score. The very first line describes the results obtained by the "duplicate" baseline, which was included to highlight and contrast the baseline performance with the presented models. In general, we can conclude that all of the proposed models managed to beat the baseline when considering the average performance across all languages, in some cases by a significant margin. The zero-shot approach, denoted as ”LLama 3” in Table 2 obtained average performance of 0.380. Extending it with Retrieval Augmented Generation (RAG) has improved the average performance to 0.403. We then noticed that when retrieving the examples to include inside the prompt, which is an important part of the RAG pipeline, the resulting examples might be in a language different from the output language. This has the potential of confusing the model during the generation process and hence we limited the examples to only include those that are in the output language (” + select”).

We further hypothesized that instructing the model to ensure specific toxic words were not present in the final output might have a positive impact on its quality. We hence implemented an approach in which all the toxic words we obtained from the `textdetox/multilingual_toxic_lexicon` dataset were included in the prompt if they could be found in the toxic sentence to be detoxified. We further added 5 more toxic words to the prompt to allow the model to see more samples of toxic words to remove from the input. This yielded the performance of 0.420. We further experimented with the order of examples in the prompt and found that reversing their order (e.g. making sure the closese example from the validation set is mentioned the last in the prompt) has had positive impact on performance (which we denote as ”+ reverse” in Table 2. In order to see what impact does the embedding part of RAG have on the final preformance we also experiment with the embedding model and make use of the `multilingual-e5-large` model which obtained state-of-the-art performance on many retrieval

Table 2
Performance metrics across different languages for various models and their components, evaluated as part of automatic evaluation. The best performance per language is boldfaced.

Model	Average	en	es	de	zh	ar	hi	uk	ru	am
baseline	0.126	0.061	0.090	0.287	0.069	0.294	0.035	0.032	0.048	0.217
Llama 3	0.380	0.525	0.448	0.530	0.161	0.488	0.185	0.507	0.461	0.112
+ RAG	0.403	0.527	0.483	0.576	0.152	0.483	0.176	0.534	0.504	0.193
+ select	0.409	0.532	0.488	0.577	0.152	0.519	0.188	0.561	0.517	0.146
+ lexicon	0.418	**0.543**	0.497	0.575	0.160	0.536	0.185	0.602	0.529	0.135
+ reverse	0.420	0.527	**0.499**	0.563	**0.169**	0.538	**0.193**	0.602	0.523	0.167
+ multiling	0.424	0.537	0.492	**0.577**	0.156	**0.547**	0.181	**0.615**	**0.540**	0.173
+ am delete	**0.437**	0.537	0.492	0.577	0.156	**0.547**	0.181	**0.615**	**0.540**	**0.287**

tasks in the multilingual setup [11]. As the results in Table 2 suggest, it also had a positive impact in our case, albeit a relatively small one. Finally, comparing the per-language performance we noticed that the performance is steadily improving for all languages except for Amharic (am). We hence decided to re-implement the delete baseline for this langauge, which was supposed to bring its performance to about 0.270. Our repimplementation has brought the performance to 0.278 as depicted in the "+ am delete" line.

As Table 3 shows, our final model has done relatively well in many languages in manual evaluation – in English its performance was tied with human references whereas for German it even surpassed the human references. We do note, however, that the model has performed surprising poorly for Hindi (hi), suggesting that there is room for potential improvement in the future, for instance by updating the language-specific prompts in this particular case.

Table 3
Performance metrics across different languages for various models and their components, evaluated as part of the manual evaluation of a random subsample of 100 texts via crowdsourcing.

Model	Average	en	es	de	zh	ar	hi	uk	ru	am
human reference	0.85	0.88	0.79	0.71	0.93	0.82	0.97	0.90	0.80	0.85
ours	0.71	0.88	0.71	0.85	0.68	0.78	0.52	0.63	0.65	0.69

7. Conclusion

In this study, we assess how well the Llama3 model performs when enhanced with retrieval augmented generation and a lexicon of toxic words for the purpose of text detoxification. We investigate the effectiveness of utilizing the Llama3 model in this context. By populating the prompt with pairs of toxic and neutral text examples, as well as toxic words from the lexicon, we achieve promising results. Our approach ranks 5th in manual evaluations of the results, with the 4th best approach having the same score and the best approach being better by only 0.06, indicating competitive performance. Particularly noteworthy is the strong performance of languages such as German and Chinese, in which our approach emerged as the second best performer.

Acknowledgements

This research was partially supported by grant APVV-21-0114.

References

[1] D. Dementieva, D. Moskovskiy, V. Logacheva, D. Dale, O. Kozlova, N. Semenov, A. Panchenko, Methods for detoxification of texts for the russian language, Multimodal Technologies and Interaction 5 (2021). URL: https://www.mdpi.com/2414-4088/5/9/54. doi:10.3390/mti5090054.

[2] D. Dale, A. Voronov, D. Dementieva, V. Logacheva, O. Kozlova, N. Semenov, A. Panchenko, Text detoxification using large pre-trained neural models, in: M.-F. Moens, X. Huang, L. Specia, S. W.-t. Yih (Eds.), Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Online and Punta Cana, Dominican Republic, 2021, pp. 7979–7996. URL: https://aclanthology.org/2021.emnlp-main.629. doi:10.18653/v1/2021.emnlp-main.629.

[3] V. Logacheva, D. Dementieva, S. Ustyantsev, D. Moskovskiy, D. Dale, I. Krotova, N. Semenov, A. Panchenko, Paradetox: Detoxification with parallel data, 2022, pp. 6804–6818. doi:10.18653/v1/2022.acl-long.469.

[4] D. Moskovskiy, D. Dementieva, A. Panchenko, Exploring cross-lingual text detoxification with large multilingual language models, in: S. Louvan, A. Madotto, B. Madureira (Eds.), Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop, ACL 2022, Dublin, Ireland, May 22-27, 2022, Association for Computational Linguistics, 2022, pp. 346–354. URL: https://aclanthology.org/2022.acl-srw.26.

[5] D. Dementieva, V. Logacheva, I. Nikishina, A. Fenogenova, D. Dale, I. Krotova, N. Semenov, T. Shavrina, A. Panchenko, Russe-2022: Findings of the first russian detoxification shared task based on parallel corpora, 2022, pp. 114–131. doi:10.28995/2075-7182-2022-21-114-131.

[6] V. Logacheva, D. Dementieva, I. Krotova, A. Fenogenova, I. Nikishina, T. Shavrina, A. Panchenko, A study on manual and automatic evaluation for text style transfer: The case of detoxification, Proceedings of the 2nd Workshop on Human Evaluation of NLP Systems (HumEval) (2022). URL: https://api.semanticscholar.org/CorpusID:248780050.

[7] TextDetox, Multilingual toxic lexicon, https://huggingface.co/datasets/textdetox/multilingual_toxic_lexicon, 2023. Accessed: 2024-05-30.

[8] TextDetox, Multilingual paradetox, https://huggingface.co/datasets/textdetox/multilingual_paradetox, 2023. Accessed: 2024-05-30.

[9] D. Dementieva, D. Moskovskiy, N. Babakov, A. A. Ayele, N. Rizwan, F. Schneider, X. Wang, S. M. Yimam, D. Ustalov, E. Stakovskii, A. Smirnova, A. Elnagar, A. Mukherjee, A. Panchenko, Overview of the multilingual text detoxification task at pan 2024, in: G. Faggioli, N. Ferro, P. Galuščáková, A. G. S. de Herrera (Eds.), Working Notes of CLEF 2024 - Conference and Labs of the Evaluation Forum, CEUR-WS.org, 2024.

[10] M. Popović, chrF: character n-gram F-score for automatic MT evaluation, in: O. Bojar, R. Chatterjee, C. Federmann, B. Haddow, C. Hokamp, M. Huck, V. Logacheva, P. Pecina (Eds.), Proceedings of the Tenth Workshop on Statistical Machine Translation, Association for Computational Linguistics, Lisbon, Portugal, 2015, pp. 392–395. URL: https://aclanthology.org/W15-3049. doi:10.18653/v1/W15-3049.

[11] L. Wang, N. Yang, X. Huang, L. Yang, R. Majumder, F. Wei, Multilingual e5 text embeddings: A technical report, 2024. arXiv:2402.05672.

A. Zero Shot prompt template

```
You are a text de-toxifier. You receive a toxic text and your task is to re-
    write the toxic parts in a non-toxic way while saving the main content.
    Do not add more text, if the text is short, keep it short. Keep the non-
    toxic words in the text the same. You do not respond anything else, just
     the rewritten, non-toxic text. The input is in "" and can be in any
    language. The result must be in the same language as the input. "{phrase
    }"
```

B. Few Shot with languages and examples prompt template

We haven't included the entire prompt due to its length, but there are similar examples for other languages where it says <examples for other languages>.

```
You are a text detoxifier. On input you receive a text which may be toxic
    or harmful. Your task is to rewrite this text in a non harmful or
    toxic way while preserving the original content and context.
Output only contains the detoxified text, nothing else like notes or
    additional information. You do not add any more context to the
    resulting text, which is not in the original text. Try not to rewrite
    the original text too much, just either remove the toxic part
    completely, or replace it with some non-toxic words while preserving
    the meaning and context.
The language of the input is {language} and the language of the output
    must be the same.
Here are some examples in English what should be the output for given
    texts:
- Input: "she is looking like shite .",
  Output: "she is looking like bad",
- Input: "btw jonhson county is a real shithole in case you were
    wondering .",
  Output: "Johnson county is not good.",
- Input: "why are canadians so taken up with hockey when their teams are
    all losers ?",
  Output: "Why are canadians so taken up woth hockey?",
- Input: "keep crying me a river , bitch boy .",
  Output: "keep crying me a river , ya boy .",
- Input: "until feminists take on islam and how it treats women , the
    entire movement is a total farce !",
  Output: "Feminism is not a good movement",
- Input: "countless crap has been spotted by me .",
  Output: "countless wrong things has been spotted by me .",
or in other languages:
- Input (in German): "dieser Holtmann scheint ein Grüner-Bastard zu sein!
    #BerichtAusBerlin",
  Output: "Dieser Holtmann scheint ein Grüner zu sein! #BerichtAusBerlin"
- Input (in Spanish): "pues hazme los deberes furcia barata",
  Output: "Pues hazme los deberes a tiempo barata."
<examples for other languages>

Provide only the detoxified text as the output, no additional information
    . Input (in {language}): "{phrase}"
```

C. RAG prompt

```
## Task

You are a text detoxifier. On input you receive a text which may be toxic or
    harmful. Your task is to rewrite this text in a way that does not
   contain any toxicity or harmful words, while preserving the original
   content and context.

The Output contains only the detoxified text and nothing else like notes or
   additional information. You do not add any more context to the resulting
    text, which is not in the original text.
Do not rewrite the original text too much, just either remove the toxic part
    completely, or replace it with some non-toxic words while preserving
   the meaning and context.

The language of the input is {language} and the language of the response
   must be the same.

## Toxic words

The input text may contain offensive or harmful words. You should either
   remove them or replace them with non-offensive words.

Here are some examples of toxic words you may find in the input text:

{toxic_words}

These CAN NOT be used in the output text. You must replace them with non-
   toxic words. If that is not possible, remove them completely.

## Examples

{examples}

## Input

Your output is only the detoxified text, you do not say anything else.

Input: {phrase}
Output:
```