# Final report

## About the task

Text Detoxification Task is a process of transforming the text with toxic style into the text with the same meaning but with neutral style. I need create any model, algorithm which will do it.

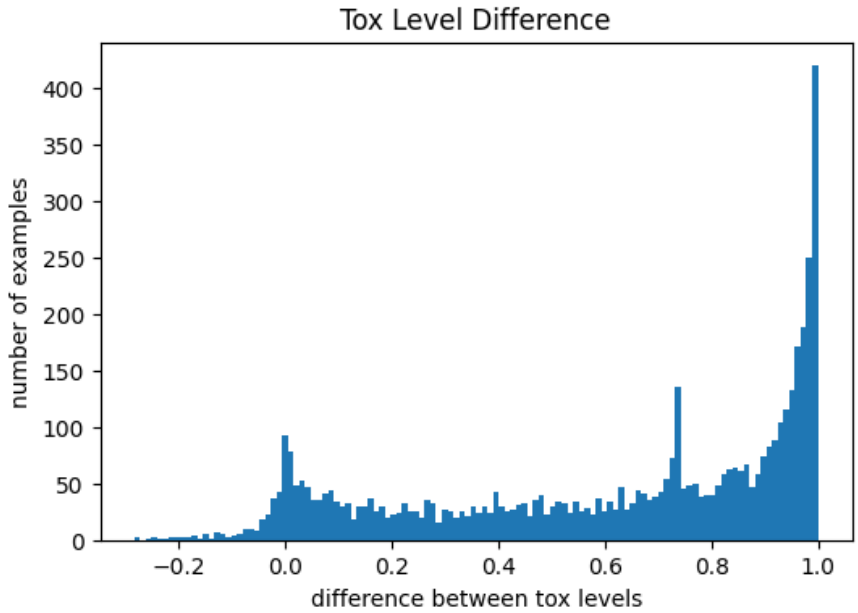
While searching for information about this problem, I found many different methods and algorithms. I decided to create a simple dictionary and use a pre-trained BERT model.

## Preparing data

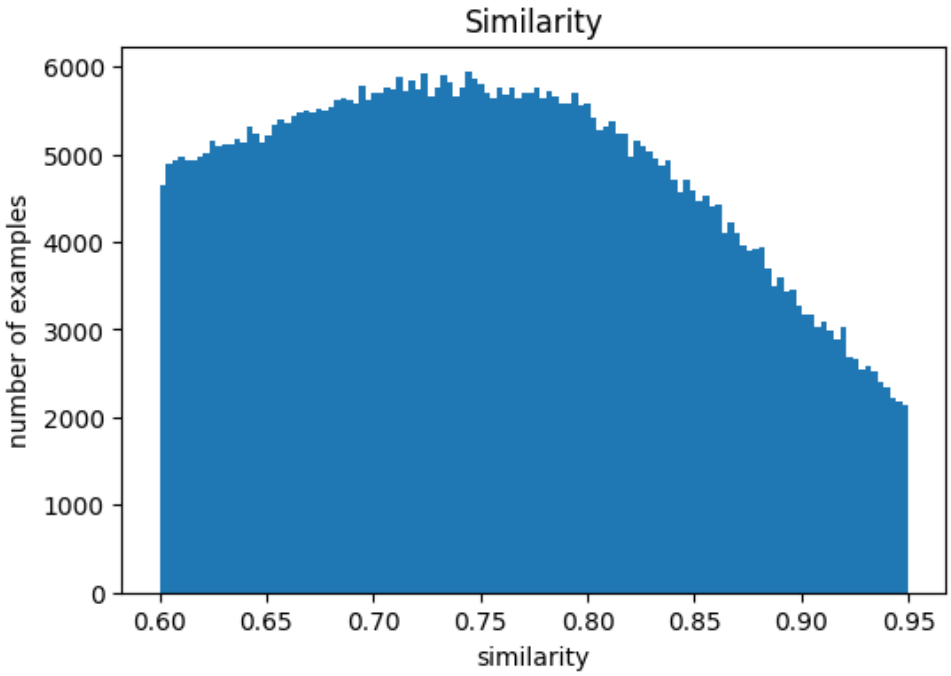
The data set that is used in this work is called ParaNMT and consists of:

- reference – text containing toxic words
- translation – the same text but toxic words replaced with normal
- toxicity level of each text and translation, as well as their similarity

To calculate whether the translation is useful, I created a new variable in the dataset 'difference', which consists of the differences in the toxicity coefficient of the reference and the translation.




Based on this graph, I selected the lines with the greatest difference in toxicity for effective training and testing. For example, in the test dataset, the toxicity level of the reference is greater than 0.8.




Since the entire dataset is very large, I decided to filter it out a bit even for training and creating dictionaries. In the training dataset, the "difference" is greater than 0.8, and the "similarity" is greater than 0.7.

But this was only the first stage of data processing. To create a "smart" dictionary, we needed to tokenize all the data and then determine the toxicity of each token, thus creating a set of good and bad words, as well as their toxicity.

## Simple solution

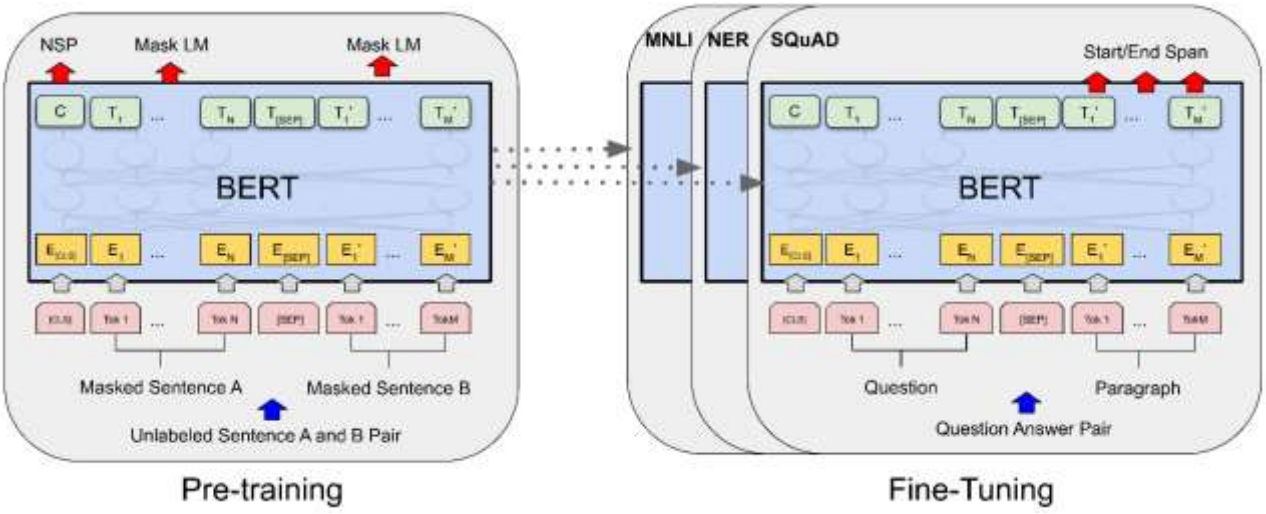
The logic of a simple dictionary is such that we find substitutions in the text and use them in the future, but in this way we do not take into account that bad words can be replaced by the same bad and toxic words, so let's take a closer look at the improved version of the dictionary and its application in the condBERT model.

Also, in the process of writing this dictionary, I will create a model that predicts the level of toxicity of the text. I trained her on both toxic and normal texts as a result, the model predicts the level of toxicity with an accuracy of 0.74.

## Model

BERT (Bidirectional Encoder Representations from Transformers) is a recent paper published by researchers at Google AI Language. For my solution I used BERT-

based model which identifies toxic words in a text and replaces them with neutral synonyms.




CondBert is designed to work with pre-trained BERT models and contains methods for identifying and masking toxic or non-toxic words in input texts. It also allows you to replace these masked tokens, offering detailed control over the rewriting process.

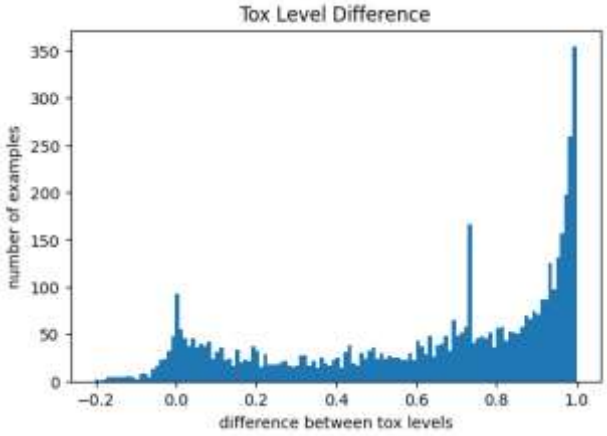
One of the key 'translate' methods takes the input text, identifies the tokens to be masked based on their toxicity, and generates a rewritten version. This method is for automatic text editing, with the ability to replace certain words or expressions with others, making it more appropriate or changing its meaning. In addition, the replacement_loop() method which allows automatically change words in the text. She searches for words that need to be replaced, suggests alternative options and makes a replacement

The model also has a get_mlm_contrast() method for calculating the contrast between different mood labels in a given text, which is useful for analyzing the sensitivity of the model to mood changes.

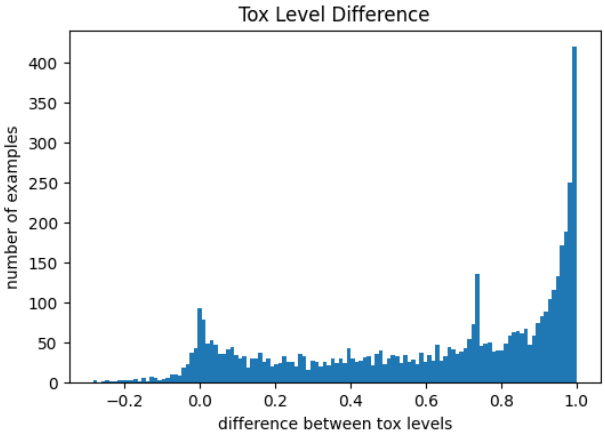
## Final result

When I realized my two hypotheses, I visualized the translations that I got. The difference in the graphs is insignificant, but ...

After studying the contents of the translations themselves in more detail, I noticed that when a simple dictionary translate bad, condBERT – well and vice versa.

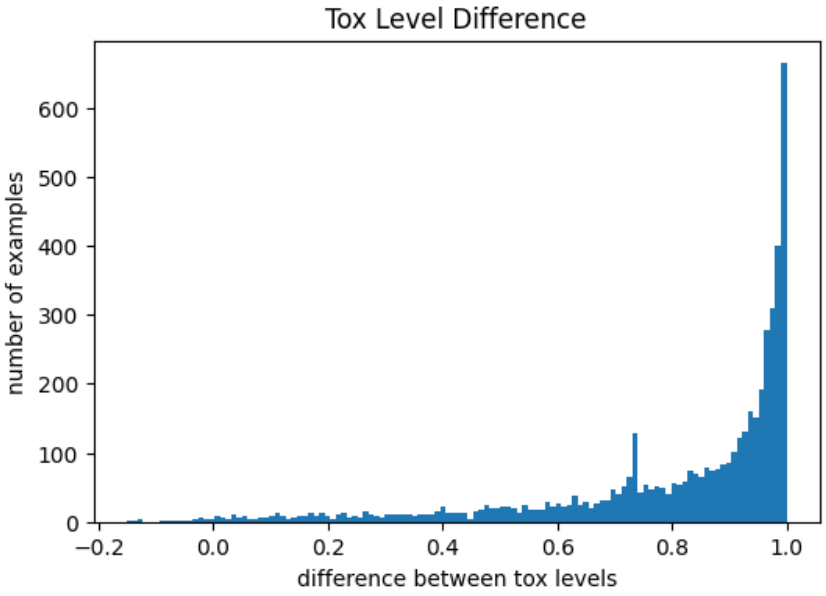



(simple vocab)



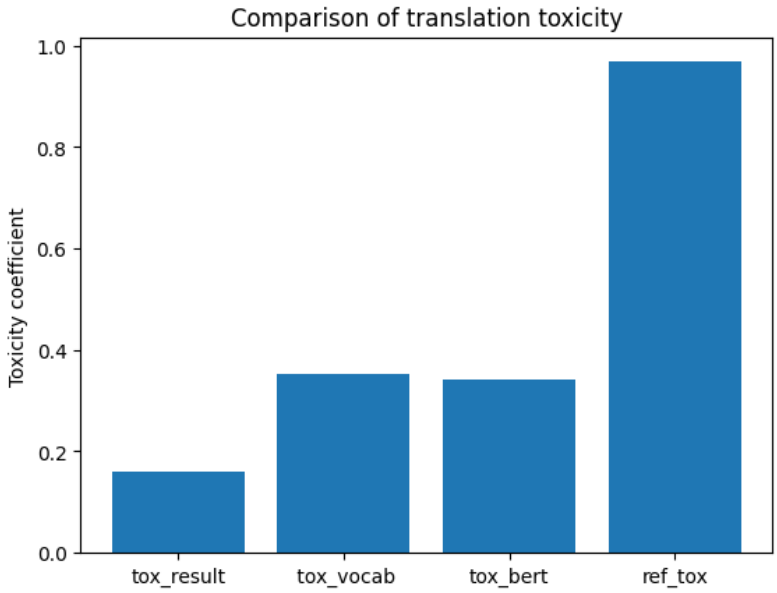

(condBERT)

Therefore, I decided not to miss the opportunity to use all my algorithms in the final solution. I combined both solutions, sorted them by toxicity level and removed duplicate rows from the combined dataset that are closer to the beginning of the table. Therefore, I chose the best translations from the two models and improved the translation quality of the entire algorithm.On the graph, you can see that the quality of the translation has improved significantly.




It can be noted that this graph shows that the algorithm translates better than it was in the original dataset.

In order to have specific figures, I calculated the average toxicity value of the translation for each solution and compare it with the initial level of toxicity.




toxicity of the final translation – 0.16

toxicity using simple vocab – 0.35

toxicity of the condBERT – 0.34

reference toxicity – 0.96

Thus, using the latest mini algorithm, we have increased the efficiency of our code by 20% on average. My entire algorithm reduces the toxicity of the text by 80%.

## Improvement in future

How can this algorithm be improved?

Firstly improve the accuracy of the model for predict the level of toxicity of the text.

Then you can try to change the models for detoxification or improve the existing condBERT.