Final Report

Introduction

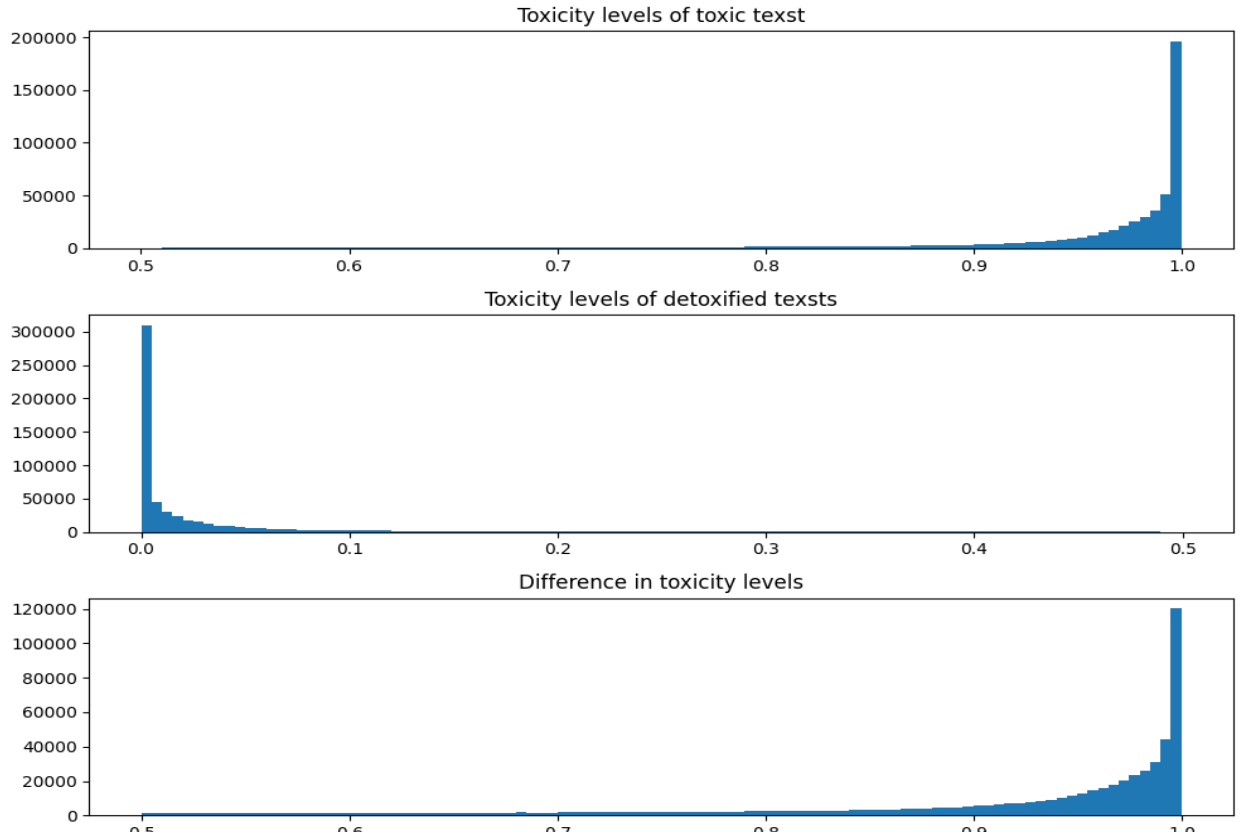
The aim of the assignment is to create a solution for the text detoxification problem. Text detoxification refers to removing toxic language from a text preserving the original meaning. The problem is important because nowadays the Internet is full of offensive and toxic language due to the opportunity to use it remaining safe. However, some individuals may find this style inappropriate and harmful. Automating the detoxification, while leaving the same meaning can enhance the quality of the Internet content, and help to filter the data for language models training.

Data Analysis

For this task the subset of 500K samples from the ParaMNT dataset was used.

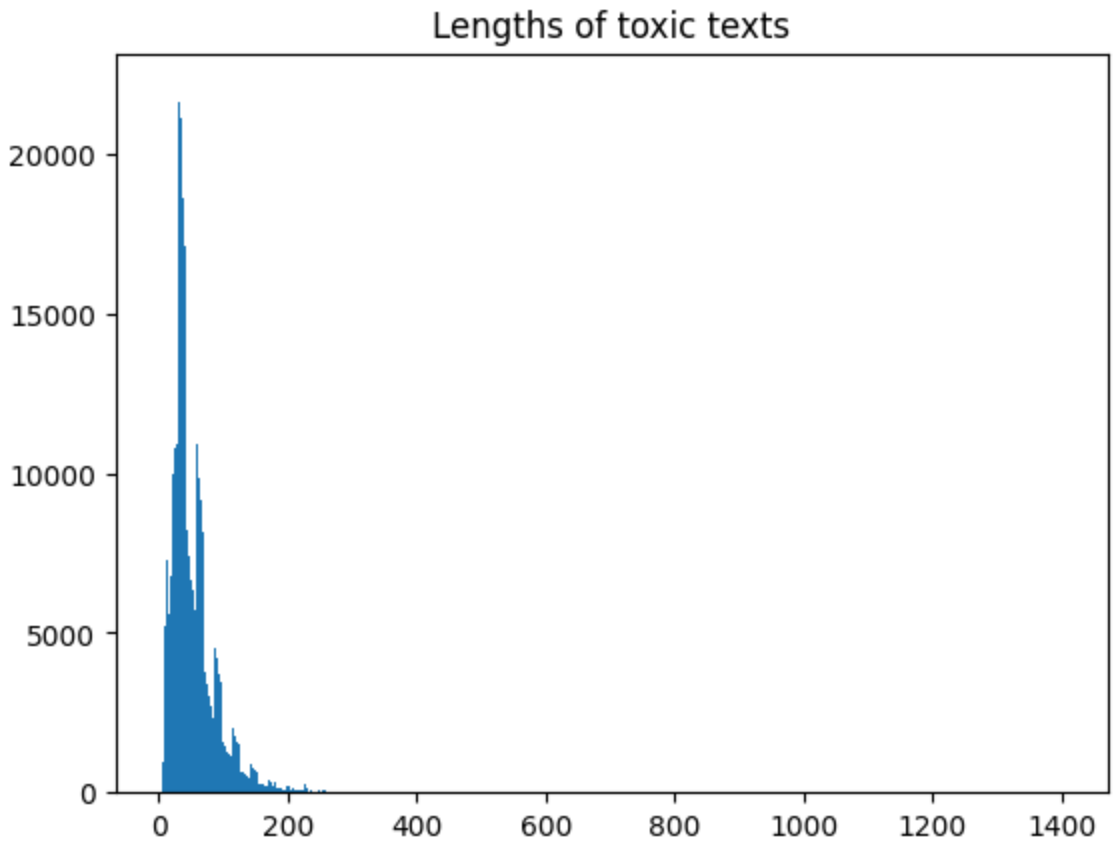
First of all, when I started exploring data, I found out that the toxic and non-toxic texts are randomly distributed between reference and translation, so I had to fix it.

Then, I examined, how the toxicity level changed after the translation



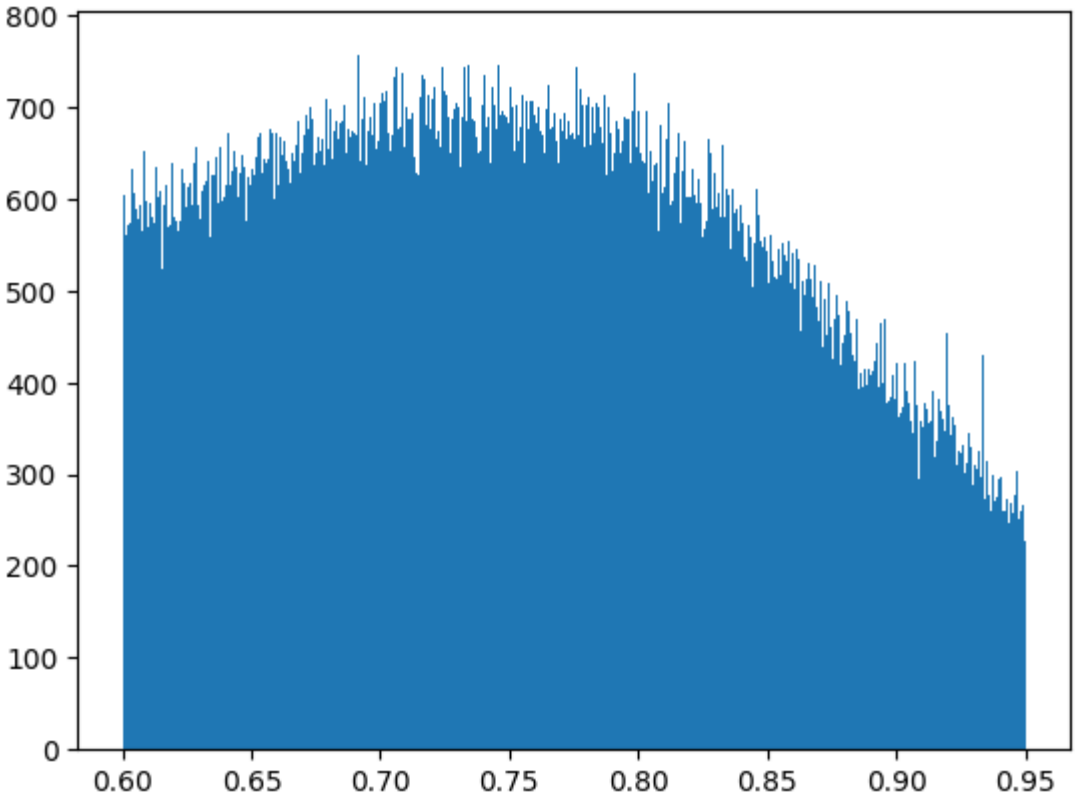
It seems that most of the sentences became less toxic after translation, but there were some outliers. To improve the results of our model, we can drop the sentences that were translated poorly.

The next step was to examine the lengths of original texts



The vast majority of the texts are shorter than 200 symbols, so it may be reasonable to drop the large outliers.

The similarity between reference and translation seemed to be distributed more or less evenly in the dataset, so it did not give me any ideas on how can I improve my search



Also, I estimated the correlation between the given features and toxicity level decrease, however, it was low, so, I concluded that it did not affect the results.

Then, I estimated what words are frequently met in the toxic dataset, but not in detoxified. Some of the words were much more often in the toxic dataset, so it gave me an idea for a baseline solution: filter out the “bad” words.

Finally, I took a look at the dataset myself, and found out that the quality of the data is actually quite low: there are many texts that were marked as non-toxic, however, for any human they would be quite offensive. I believe that the toxicity was labeled by some model, and not humans. However, it was decided to work with the dataset anyway.

Before passing the data to my model, I lowered all the texts, expanded contractions, and filtered the dataset to contain only english words and numbers.

Model specification

For the task I have tried three models. The models are described in more detail in the first report. As for the model that I trained myself: it is a Transformer [4] model with classical architecture. The model includes word and positional embeddings, 8 attention heads, 3

encoder and decoder layers, 4 as a generator dimension. The maximum sentence size is 128. The size of vocabulary was limited to 25 000 most common words. The architecture and code were inspired by the tutorial [5]

Training process

I trained the model using CrossEntropyLoss ignoring the padding index. As an optimizer, I used Adam with a learning rate 0.0003. The model was trained for 25 epochs, but started overfitting after 10-15, so the checkpoint with the smallest validation loss was used for inference. The batch size was 64. The best validation loss was about 2.13.
I used a kaggle environment with a P100 GPU for training the model, and the overall training time was 7.5 hours.
During the training process I encountered different challenges. One of them was that I had to retrain the model at least four times: first time was unsuccessful due to poorly chosen hyperparameters, second one failed due to hardware-related issues, and the third attempt was ruined by my mistake. Due to the large training time, I faced the lack of free hardware capable of training my model fast and effectively. Also, the data itself contained a lot of different words, so I had to limit my vocabulary size to 25000 to speed up the computations.

Evaluation

To evaluate my model I needed metrics to measure the semantic similarity of the original and translated texts, to measure the toxicity level, and to find out how similar sentences are produced to expected results. For similarity I used a pretrained model [6] that produces embeddings for sentences. For toxicity I used a pretrained model[7] that measures toxicity level from 0 to 1. To determine, how close are my predictions to the target result, I used the following metrics:

BLEU score: Bilingual Evaluation Understudy. It uses n-grams to compute the similarity between two sentences. The problem with this metric is that it produces a result of 0 for short sentences, so I did not take into account the cases when the score is 0.

ROUGE score: Recall-Oriented Understudy for Gisting Evaluation. More specifically, I used ROUGE1 and ROUGE2 with an F1 score. These metrics consider the F1 score of matching unigrams and bigrams of the sentences correspondingly.

Due to high computational complexity of my models and some metrics, I limited the test set to 1000 random samples from the provided ParaNMT dataset.

The results:

Model name	BLEU (zeros excluded)	ROUGE1 F1	ROUGE2 F1	Toxicity	Semantic Similarity with original sentences
Baseline	0.3615	0.5722	0.3318	0.3154	0.8724
Pretrained T5	0.4175	0.5558	0.3023	0.3346	0.8535
Transformer	0.5269	0.6175	0.4051	0.1603	0.7668

The results show that all of the models are quite successful in the task. They can produce the texts that are similar to the original sentence by the meaning, but have lower toxicity. For examples of work you can take a look at the data/interim/testing section.

On the other hand, the models are still not good enough compared to people that can do the same task. To further improve quality, one can use larger and wider datasets with better data quality or more complex models could be used. However, to reach a human level some more extensive research into style transferring and content preservation could be used.

Summing up, the results of the assignment show that neural networks have a potential in the text detoxification task, but some more research may be needed. Improving the quality of detoxification algorithms may help to create a better user experience for Internet users and help LLM developers to prepare data for training.