Report on Solution Creation

Leila Khaertdinova, BS21 DS-02

November 2023

Baseline: Delete

In this report, I will discuss and outline the process steps I followed to create a solution for the task of text detoxification.

This baseline approach that firstly comes to my mind draws inspiration from how offensive language is handled in TV programs and articles, involves substituting swear words with asterisks (*). Following a similar principle, goal of this baseline is to just delete offensive words from the source text with a high toxicity level. To implement this approach, a dictionary was created, consisting of the bad words obtained from the obscenity list from the Github repository mentioned in the references directory.

Example:

Source sentence	**Detoxified sentence**
"What a stupid joke."	"What a joke."
"Fucking damn joke!"	"joke!"

Table 1: Baseline results

As can be seen from the table above, the bad words from the example source sentences were removed. However, this approach is not the perfect one as it may result in the loss of contextual information within the original toxic sentence.

Hypothesis 1: Pytorch transfromer

As an alternative to the baseline delete approach, I explored the use of a Seq2Seq models for solving machine translation tasks. However, they also can be applied for text detoxification.

The hypothesis behind this approach is that by leveraging the power of a transformer model, we can replace offensive language style with the same meaning but in a neutral manner. In this step, I implemented the Seq2Seq transformer from scratch using Pytorch.

Example:

Source sentence	Detoxified sentence
"What a stupid joke."	"what a joke."
"Fucking damn joke!"	"you got ta be kidding me."

Table 2: Pytorch transformer results

Consequently, the Pytorch transformer provides quite good results in a non-toxic manner, but some information from the original example sentences is missing. As shown in the table, the detoxified sentences are less expressive than the original examples. Therefore, further improvements are needed to make the detoxified sentences more informative and natural-sounding.

Hypothesis 2: Fine-tuning T5 model

For this step, I explored the fine-tuning process for the given task. I used a pre-trained T5 model with T5-Base checkpoint taken from the HuggingFace and fine-tune it on a preprocessed dataset that mainly used for our text detoxification task.

Example:

Source sentence	Detoxified sentence
"What a stupid joke."	"what a bad joke."
"Fucking damn joke!"	"it's a terrible joke!"

Table 3: T5 fine-tuned results

Based on the these examples, it can be concluded that the model can detoxify toxic sentences and provide non-toxic alternatives in a natural way.

Results

In this report, different approaches for the task of text detoxification were explored. I started with a baseline approach that involved deleting offensive words from the source text. While this approach successfully removed the offensive words, it also resulted in the loss of contextual information.

Then, I explored the use of Seq2Seq models, specifically the Pytorch transformer model. The examples showed that the Pytorch transformer model provided good detoxification results, but the detoxified sentences were less expressive compared to the original examples.

Finally, I fine-tuned T5-base model on the main detoxification dataset. Based on provided examples, the fine-tuned T5 model successfully detoxified the toxic sentences and provided non-toxic alternatives that sounded more natural.

The details about models training, evaluation and visualizations will be provided in the final report.