Solution Building Report

1. Review and processing of source data

Our initial dataset has the following structure:

	reference	translation	similarity	lenght_diff	ref_tox	trn_tox
0	If Alkar is flooding her with psychic waste, t...	if Alkar floods her with her mental waste, it ...	0.785171	0.010309	0.014195	0.981983
1	Now you're getting nasty.	you're becoming disgusting.	0.749687	0.071429	0.065473	0.999039
2	Well, we could spare your life, for one.	well, we can spare your life.	0.919051	0.268293	0.213313	0.985068

As we can see, in our dataset, sentences are simply translated into their opposite toxicity levels. This implies that a sentence deemed toxic in the ‘reference’ column becomes non-toxic in the ‘translation’ column, and vice versa. Given our objective to segregate all toxic sentences into one column and non-toxic sentences into another, our initial logical step involves reorganizing all toxic sentences into the ‘reference’ column and all non-toxic sentences into the ‘translation’ column. We’ll categorize a sentence as toxic if its toxicity score is greater than 0.5. Save new file in .csv file. Also we have ‘lenght_diff’ and ‘ref_tox’ columns, maybe they can be used for future processing.

2. Logistic regression for word toxicities scores

The paper “Text Detoxification using Large Pre-trained Neural Models” by David Dale et al. offers valuable information for our initial ideas. Specifically, Chapter 4 of the paper discusses the implementation of a Conditional BERT Model.

Initially, our aim is to develop a classifier, specifically a logistic regression model, that categorizes sentences as either toxic or neutral based on the words they contain, which serve as features. An intriguing outcome of the training process is that each feature (word) is assigned a weight that approximately signifies its relevance to classification. Here’s how our dictionary appears:

```
'abarrach': 0.157464281969969,
'abarrachuk': -0.06503823860114374,
'abbey': -0.16188603512028638,
'abbot': -0.13446714795453282,
'abby': 0.3177238742164888,
'abdo': -0.14904427980630758,
```

This method proves to be especially beneficial for our text detoxification task as it enables us to measure the toxicity of individual words within a sentence. By substituting or altering the words with high toxicity weights, we can decrease the overall toxicity of a sentence while maintaining its original meaning. This is consistent with our objective of converting highly toxic text into text that conveys the same meaning but in a neutral tone.

3. Initializing toxic words according to their toxicity scores

In the process of text detoxification, a key step is to compute the toxicity score for each word in a sentence and define toxic words as those with a score above a certain threshold. This threshold is calculated as the maximum of two values: a minimum toxicity score (tmin = 0.2) and half of the maximum toxicity score among all words in the sentence. This is example of our toxic words:

```
'fatass',
'ejaculation',
'crafted',
'goddamit',
```

This adaptive thresholding technique allows us to balance the percentage of toxic words in a sentence, thereby avoiding cases where too many or no words are marked as toxic. By applying this approach, we can effectively identify and replace or modify toxic words, reducing the overall toxicity of a sentence while preserving its original meaning.

4. Simple BERT model for predict masking toxic words

We currently try to develop an algorithm to censor offensive words in text. Firstly, we replace each inappropriate term with a placeholder, [MASK]. Then we utilize BERT, a pre-trained language model, to censor offensive words in text. By loading a pre-trained BERT model and tokenizer, we can tokenize the test text and predict the masked tokens. This approach allows us to effectively mask toxic words in our text data.

The BERT model is then able to predict these masked words based on their context.

5. Results

We have gathered several initial ideas for implementing our final solution, and we anticipate incorporating many of them. However, we recognize the need to refine and optimize some approaches or explore additional strategies. It is worth mentioning that the performance of the basic BERT model in predicting masked words has shown some variability and falls short of the desired accuracy at this stage. To achieve a more reliable and effective solution, we want either developing a more advanced model or leveraging a high-performing model from open-source repositories. This will help ensure the desired level of accuracy and improve the overall effectiveness of our solution.