Final Solution Report

1. Introduction

Our task is to detoxify text, transforming any potentially harmful or offensive content into a more positive and respectful form. We will be drawing inspiration from a research paper titled “Text Detoxification using Large Pre-trained Neural Models”. This paper, particularly Chapter 4, provides valuable insights into the use of the Conditional BERT Model for our purpose.

In addition to the paper, we will also utilize an open-source model available on GitHub(https://github.com/s-nlp/detox). This model is specifically designed for rewriting toxic words into non-toxic alternatives, making it an excellent resource for our task.

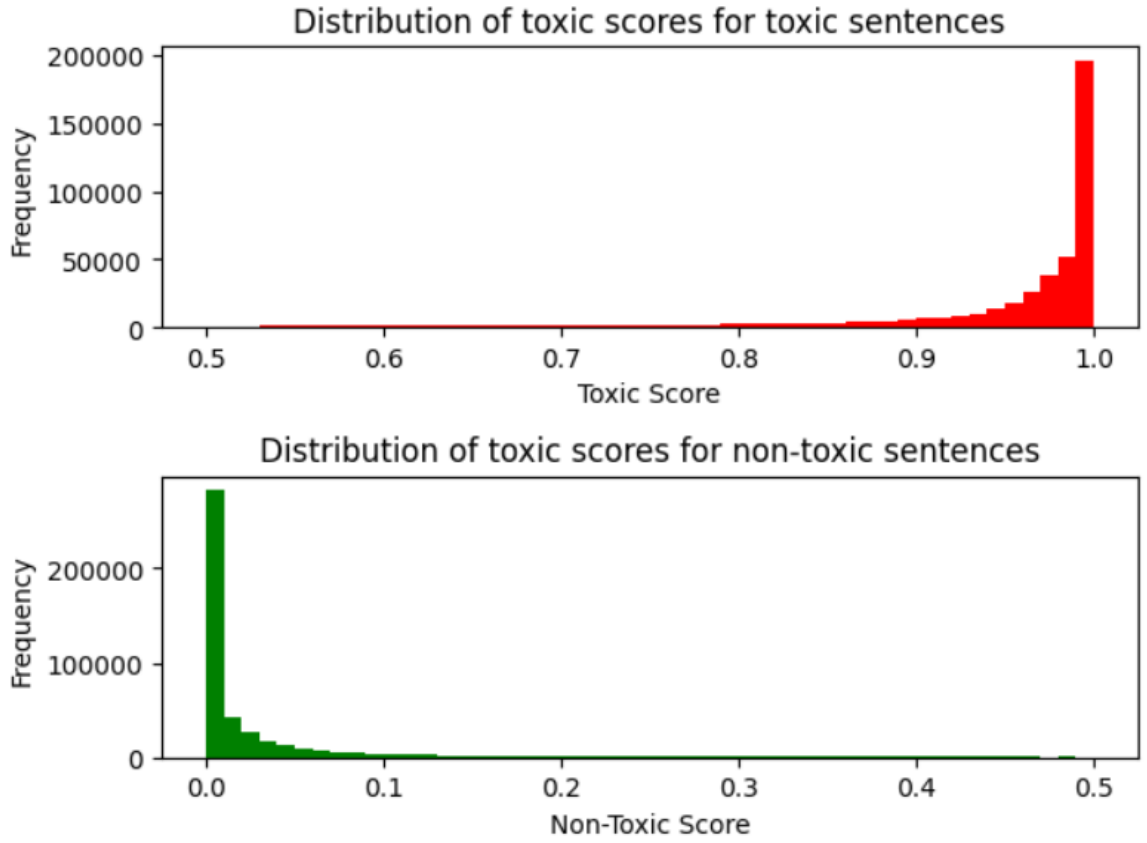
By combining the theoretical knowledge from these papers with the practical application of the open-source model, we aim to effectively detoxify text.

2. Data analysis

Let’s delve into our dataset, which we assembled in the preceding notebook. We’ll find that all toxic sentences have been placed in the “reference” column, while their paraphrased, non-toxic counterparts reside in the “translation” column. Here’s a glimpse at a few columns from our dataset:

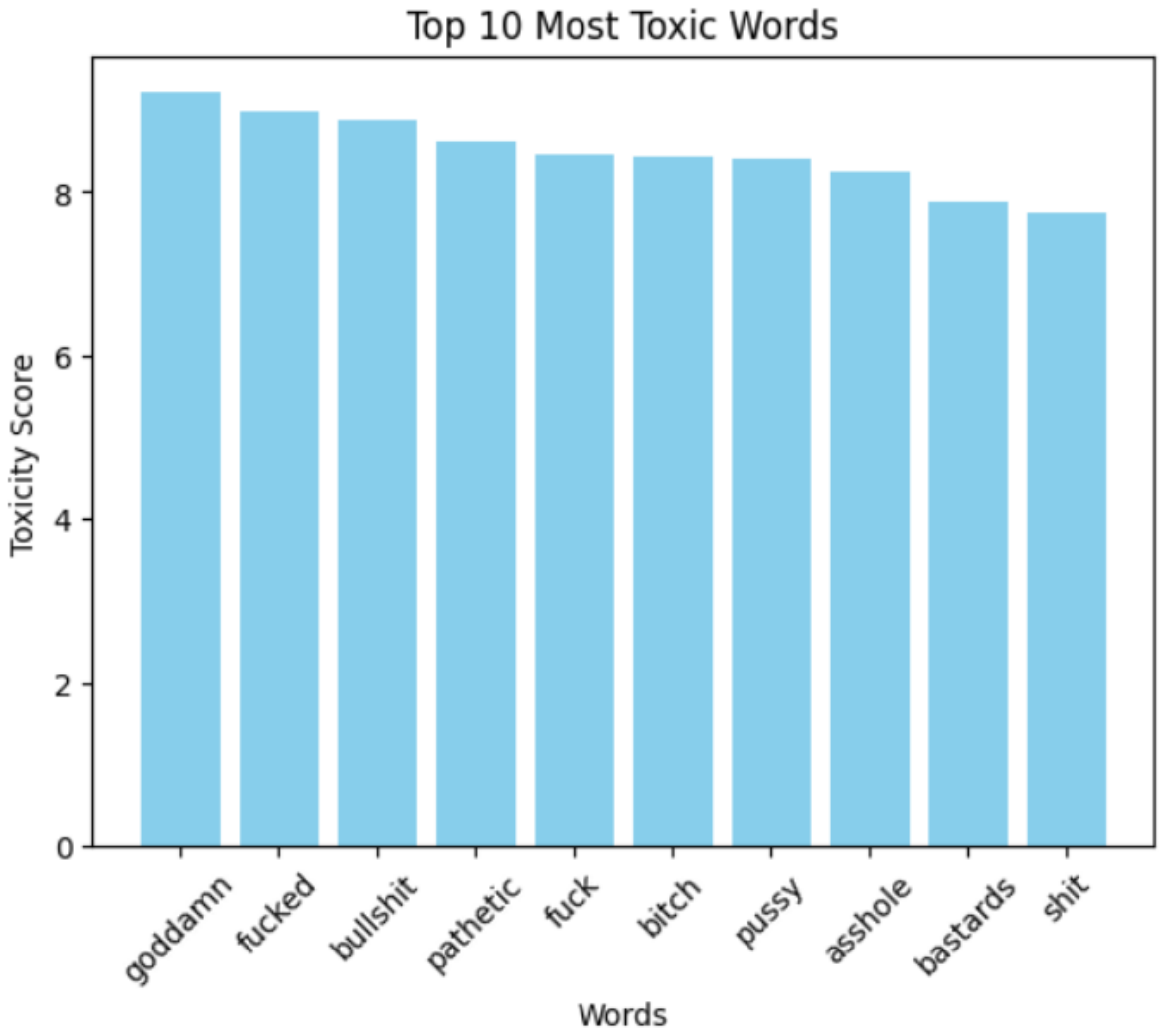
	reference	translation	similarity	lenght_diff	ref_tox	trn_tox
0	if Alkar floods her with her mental waste, it ...	If Alkar is flooding her with psychic waste, t...	0.785171	0.010309	0.981983	0.014195
1	you're becoming disgusting.	Now you're getting nasty.	0.749687	0.071429	0.999039	0.065473
2	well, we can spare your life.	Well, we could spare your life, for one.	0.919051	0.268293	0.985068	0.213313
3	monkey, you have to wake up.	Ah! Monkey, you've got to snap out of it.	0.664333	0.309524	0.994215	0.053362
4	I have orders to kill her.	I've got orders to put her down.	0.726639	0.181818	0.999348	0.009402
...	...	...	...	...	...	...
577772	you didn't know that Estelle stole your fish f...	You didn't know that Estelle had stolen some f...	0.870322	0.030769	0.949143	0.000121
577773	It'il suck the life out of you!	you'd be sucked out of your life!	0.722897	0.058824	0.996124	0.215794
577774	I can't fuckin' take that, bruv.	I really can't take this.	0.617511	0.212121	0.984538	0.000049
577775	They called me a fucking hero. The truth is I ...	they said I was a hero, but I didn't care.	0.679613	0.358209	0.991945	0.000124
577776	I didn't fuck him.	I did not screw him.	0.868475	0.095238	0.994174	0.009480

Here are the visual representations of the toxicity levels distribution for sentences, as determined by people through binary classification:



Next, we'll segregate the training data into two lists: toxic_sentence for all potential toxic sentences, and nontoxic_sentence for their non-toxic counterparts. We'll assign a label of one to all toxic sentences and a label of zero to all non-toxic sentences. Using a regression model, similar to the ones we've seen in the previous notebook, we'll then determine the toxicity scores for each word.

Here, we present a graph showcasing the top 10 most toxic words along with their respective toxicity scores:

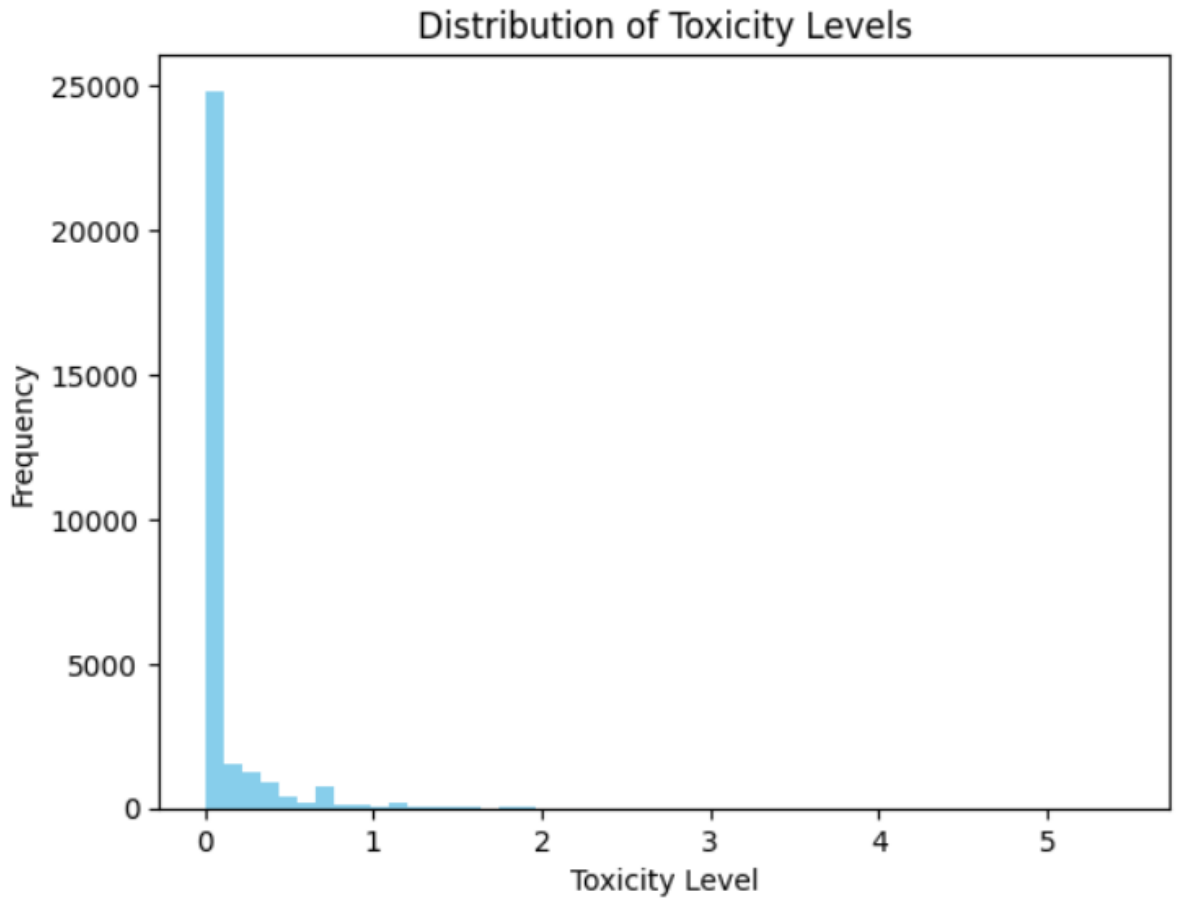


We're now enhancing our toxic words identifier to simultaneously detect both toxic and positive words. While the toxic word check continues against its threshold, we're introducing an additional threshold for identifying positive words. This is calculated as the lesser of two values: a minimum toxicity score (tmin = -0.2) and half of the minimum toxicity score among all words in the sentence. In future, our model will require the identification of both these groups of words. This is example of our toxic and positive words:

```
'smartass','month',
'hippy',   'karma',
'boner',   'urges',
'hick',    'economic'
```

We're working on figuring out how often certain words show up in text data that's considered toxic. This helps us see if there's a link between specific words and harmful language. We keep track of how many times

each word pops up in both toxic and non-toxic sentences. Then, for each word, we calculate something called a “toxicity ratio”. This tells us how often a word shows up in toxic sentences compared to all the times it shows up. After that, we change these toxicity ratios into another measure called “log-odds ratios” by making sure they fall between the maximum of 0 and the natural logarithm of the toxicity ratios divided by 1 minus the toxicity ratios. In the end, we try to avoid tokens that don’t really mean anything. This is example toxicity levels distribution for log_odds_ratios:



3. **CondBertRewriter** model specification

We plan to use an open-source model called CondBertRewriter for our goals. This tool is designed to replace toxic words in sentences with non-toxic alternatives. Here is what our model consists:

- Setting Up: We start by setting up the CondBertRewriter with a few things. This includes a pre-trained model, a tokenizer (both from BERT), the device we're running the model on, lists of toxic and

non-toxic words, a dictionary that links words to their toxicity scores, an array of toxicity ratios for each token, and a predictor that generates replacement tokens.

- Grouping Tokens: The group_by_first_token function groups sequences of tokens (words made up of multiple parts) based on their first token. This helps us handle complex words that are considered either toxic or non-toxic.
- Converting Tokens to Words: The toks_to_words function changes sequences of tokens back into whole words. This is useful when we're calculating toxicity ratios.
- Creating Masks: The get_mask_fast function creates masks for sentences based on how toxic the words they contain are. If a sentence doesn't contain any toxic words, the function just gives us back the original sentence.
- Rewriting Sentences: The translate method in CondBertRewriter rewrites a sentence by repeatedly masking spans of tokens and replacing them with new tokens generated by the predictor.

4. CondBertRewriter model training and prediction process

In our training procedure for the CondBertRewriter model, we start by checking if a GPU is available for training. If not, we use the CPU instead. We then define a function called adjust_logits that adjusts the raw prediction values (logits) output by the BERT model based on our calculated log-odds ratios, biasing the model's predictions towards less toxic tokens. Following this, we set up a MaskedTokenPredictorBert instance with our pre-trained BERT model, tokenizer, and other parameters. This predictor will generate replacement tokens for the masked spans in our sentences. Finally, we initialize an instance of CondBertRewriter with our pre-trained BERT model, tokenizer, device, lists of toxic and non-toxic words, word-toxicity scores, token toxicity ratios, and our initialized predictor. The actual "training" happens during usage when the translate method is called to rewrite sentences. The model uses its pre-trained components and learned associations (toxicity scores and ratios) to iteratively mask and replace spans of tokens in the sentences, learning to change toxic words into non-toxic alternatives.

The sentence prediction process is rather uncomplicated. We methodically input the original toxic sentences from the test set into the model. In return, it provides us with a restructured version that retains a similar meaning but is non-toxic. This examples of original toxic/nontoxic and predicted sentences:

```
["The best treatment for this is to simulate the womb Of the patient's mother using blankets Or a whale's vagina.", "I'm always
afraid to tell Véro I fucked her.", 'You are shitting me?!', "Don't be ridiculous.", "I'd like to exterminate this conversatio
n. What are you doing?", 'are you sitting on your butt?', "'Where you from?' said the troll.", 'So how many rooms am I checked
into in this shit hole?', "killing us won't bring your boy back.", 'but the fact was that Steve Kemp, published by the poet, a
nomadic repairman of furniture, a producer of knitted chairs, a good amateur tennis player and an excellent afternoon lover, wa
s - a swine.']
[["the best treatment is to simulate the womb of a patient's mother... using blankets or whale cars."], ["I'm always scared sh
e'll tell Véro I banged her."], ["you're kidding me."], ["don't be crazy."], ["I'd like to derail this conversation."], ['Are y
ou sitting down?'], ["'will this... are you? 'The troll asked."], ['how many rooms have I logged in here?'], ["the killing wo
n't bring your son back."], ['But when you took the bark off it, the simple fact was that Steve Kemp - publishing poet, itinera
nt furniture stripper and refinisher, chair caner, fair amateur tennis player, excellent afternoon lover - was a turd.']]
```

```
["the best treatment for this is to simulate the womb of the patient ' s mother using blankets or a whale ' s av womb.",
 "i ' m always afraid to tell vero i banged her.",
 'the are kiddinging me?!',
 "don ' t be fooled.",
 "i ' d like to endter silence this conversation. what are you doing?",
 'are you sitting on your back?',
 "' where you from? ' said the troll.",
 'so how many rooms am i checked into in this my hole?',
 "the us won ' t bring your boy back.",
 'but the fact was that steve kemp, published by the poet, a nomadic repairman of furniture, a manufacturer of knitted chairs,
a good amateur tennis player and an excellent afternoon lover, was - a swine.']
```

5. Evaluation

Initially, I attempted to use the BLEU metric for evaluation purposes. However, the results were rather disappointing:

```
{'score': 16.732198041101984,
 'counts': [736396, 325686, 159333, 78808],
 'totals': [1582706, 1464929, 1347163, 1230092],
 'precisions': [46.527655799624185,
  22.232203745027917,
  11.827299294888592,
  6.406675273069006],
 'bp': 1.0,
 'sys_len': 1582706,
 'ref_len': 1407171}
```

However, there's no need for immediate concern. Even though these results might seem bad, it's not particularly alarming for our task. Our goal is to detoxify the text, not to mimic its approach to the target sentences. The unsatisfactory results on this metric could be due to the fact that our method of detoxification differs from the one used for the target values.

As such, we want to make our own metric to assess text detoxification. We'll implement a Logistic Regression model, which is quite similar to the model we used to determine the toxicity scores of individual words. However, in this case, we'll use it to determine the toxicity score of entire

sentences. We'll train our model on the test data. Then, by using it to calculate the average toxicity of original toxic/non-toxic and predicted sentences, we can obtain the following values:

```
Average Toxicity Score for Toxic Sentences: 0.6997527525694864
Average Toxicity Score for Non-Toxic Sentences: 0.30034185818866566
Average Toxicity Score for Predicted Sentences: 0.30208453203700636
```

The final step in defining our metric will be to quantify the reduction in sentence toxicity achieved through the detoxification process, in relation to the total potential toxicity that could have been reduced. For this calculation we get:

```
Effectiveness of the detoxification: 0.9956368895469357
```

6. Results

In conclusion, we've made significant strides in detoxifying the text. The supplementary materials and code have been instrumental in achieving commendable results. We've not only enhanced the methods from the first notebook but also incorporated new techniques for various tasks.

Admittedly, there are a few areas for improvement. For instance, our rewritten text occasionally generates contextually incorrect content, returns everything in lowercase, and inaccurately uses the "'" sign. It also rewrites words that ideally should remain unchanged. However, on the whole, the algorithm performs well.

While other approaches might yield better results with this task, our current methods have proven to be effective. It's worth noting that our evaluation metric isn't perfect and could benefit from refinement. It doesn't consider the logic of the sentence itself and has a subjective definition of sentence toxicity. This could lead to potentially inaccurate assessments of our algorithm.

There are certainly other metrics and models that might be more suitable for this task. Nevertheless, we've made substantial progress with our current methods.