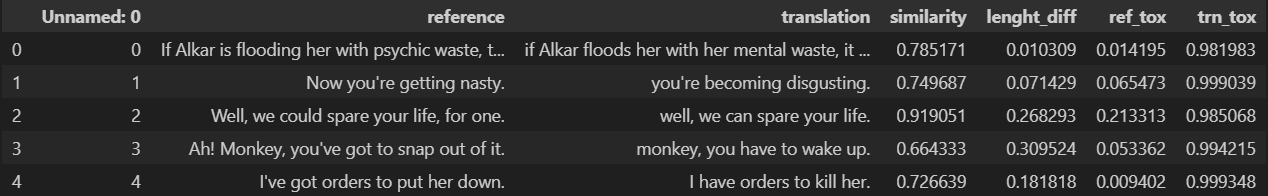
**Summary of Data Preprocessing**

1. Overall view of the data



In the given data for a text detoxification task, each column serves a specific purpose:

Unnamed: 0: This column appears to be an index for each row of data and does not provide any specific information related to the text detoxification task. It will be deleted later.

reference: This column contains the original toxic text or that needs to be detoxified.

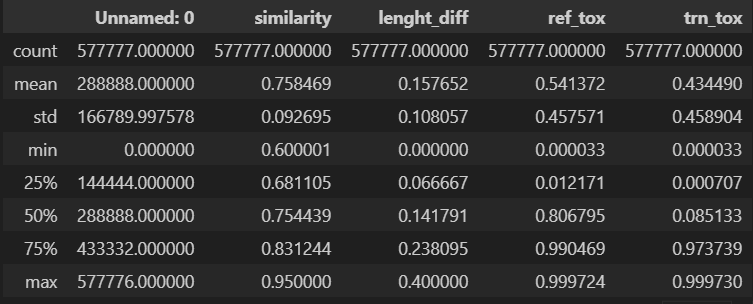
translation: This column contains the transformed text after detoxification. It represents the result of the detoxification process, where the original toxic text has been converted into a non-toxic version.

similarity: This column appears to represent a similarity score between the reference text and the translation text

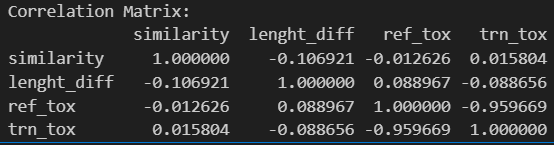
length\_diff: This column seems to capture the difference in length (number of characters or words) between the reference and translation texts.

ref\_tox: This column could represent a toxicity score or a measure of how toxic the reference text is.

trn\_tox: This column might represent a toxicity score for the translation text, indicating how toxic or non-toxic the detoxified text is.



The ref\_tox and trn\_tox values suggest that the reference texts are, on average, more toxic than the transformed texts (0.541372 vs. 0.434490). This could suggest that whatever transformation is applied to the texts tends to reduce their toxicity on average.



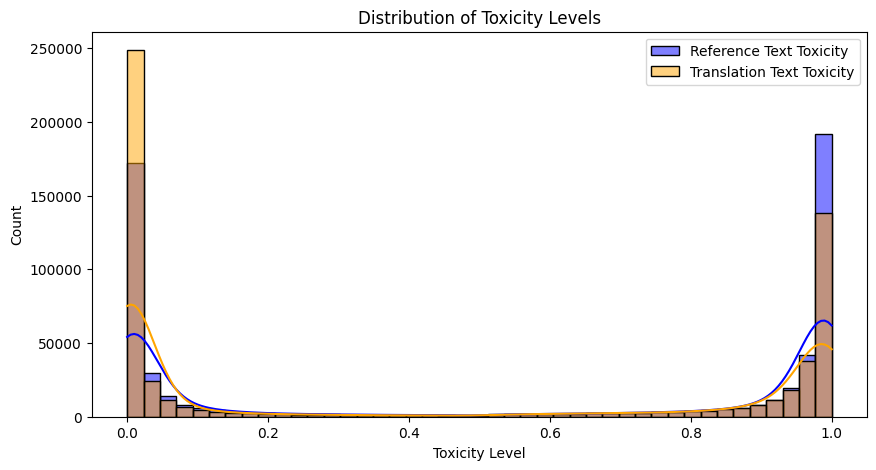
The strong negative correlation between reference and translation text toxicity suggests that, in most cases, when one text becomes more toxic, the other text becomes less toxic.

The weak correlation between length difference and any other parameters, as Ill as the weak correlation between similarity and any other parameters, indicates that this parameter is unimportant for designing the effective model.

1. Data Cleaning

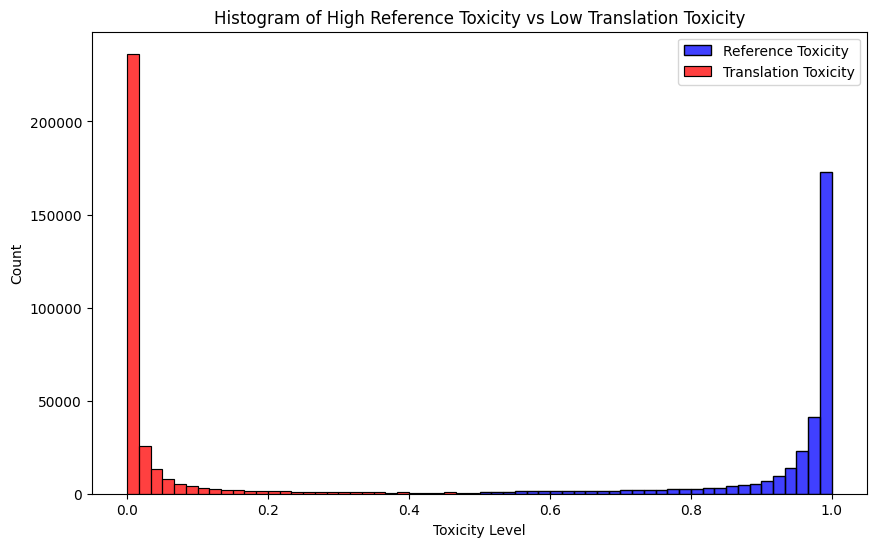
Since 'similarity' and 'lenght\_diff' are unimportant and 'Unnamed: 0' does not do anyone any good, I can drop them.

1. Dealing with toxicity distribution

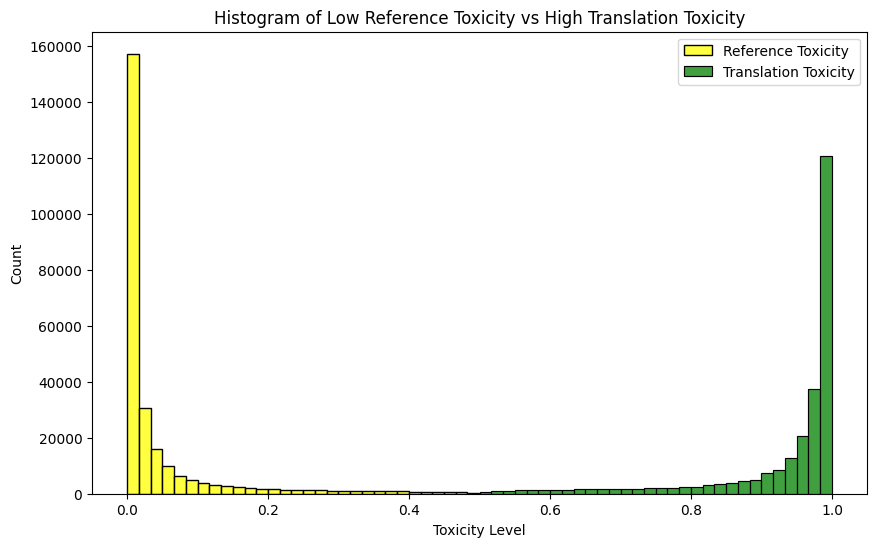


The observation that the distribution of toxicity levels is skewed to both extremes (0 and 1) indicates that the dataset contains a significant imbalance in toxicity levels. Most of the data points have toxicity values close to either 0 (non-toxic) or 1 (toxic), with few instances in between. Additionally, the distribution suggests that there may be a slight bias towards higher toxicity in translation texts and a slight bias towards lower toxicity in reference texts.

To visualize this imbalance, I have included histograms that highlight rows where reference toxicity is greater than 0.5 and translation toxicity is less than 0.5, as Ill as rows where reference toxicity is less than 0.5 and translation toxicity is greater than 0.5.

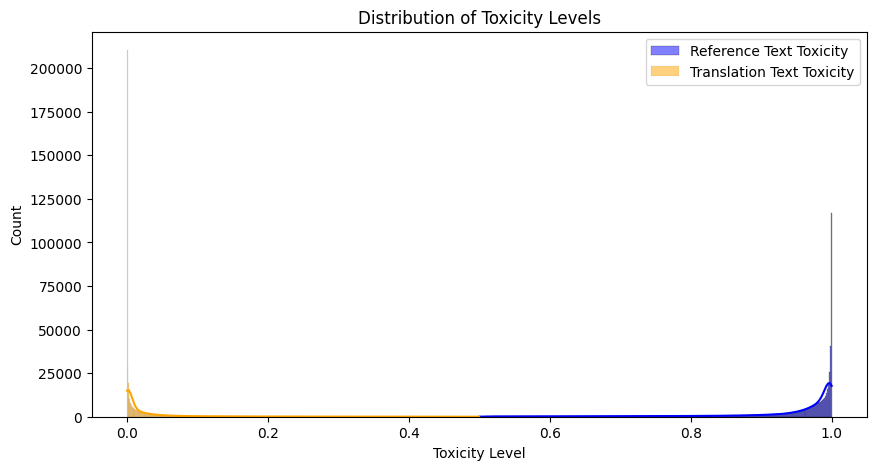


This histogram shows the rows where reference toxicity > 0.5 and translation toxicity < 0.5



This histogram shows the rows where reference toxicity < 0.5 and translation toxicity > 0.5

To address the issue of previously non-toxic sentences being transformed into toxic ones, a column swap was implemented to create a new distribution of toxicity:



Then, I deleted all rows where translation toxicity > 0.05 or reference toxicity < 0.05. I chose thresholds like 0.95 for high toxicity and 0.05 for low toxicity because these thresholds are not uncommon, especially in a binary classification scenario where I want to ensure a clear distinction betIen the "high" and "low" classes. Before deleting I had 577777 rows, after – 349705, a sufficient amount for subsequent model training.

1. Lowercasing and Removing Punctuation

To ensure uniformity and prevent the model from treating words with different casing as distinct entities, all words have been converted to lowercase. This step eliminates any potential confusion between, for example, 'Hello' and 'hello.'

Additionally, as part of the text cleaning process, extraneous characters and symbols, such as Japanese hieroglyphs, have been removed, focusing exclusively on the lowercase English alphabet. Commas and full stops have been omitted as Ill, prioritizing memory efficiency and the preservation of space for meaningful words.

1. Text Length Control:

To align with the architecture of a network with a maximum of 128 neurons in the input layer, a constraint has been imposed on the text length. Rows where either the 'reference' text or the 'translation' text exceeds 128 characters have been removed.

This data preprocessing effort resulted in the deletion of a relatively small number of rows, specifically 13,449 rows. Despite this reduction in the dataset size, it is anticipated that this streamlined and standardized data will significantly enhance training speed and model performance.

**Process Overview**

1. Initial Approach: Synonym Replacement Strategy

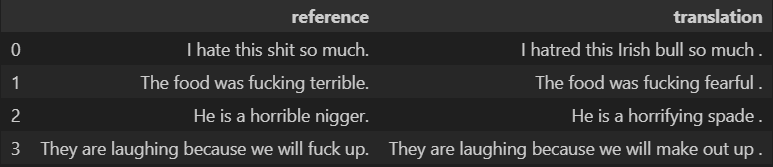
Sentiment Analysis: I started with sentiment analysis to gauge the overall sentiment of the sentences and highlight the negative words.

Tokenization & POS Tagging: I tokenized the sentences into individual words and tagged them for their part of speech to ensure synonym replacements would fit grammatically.

Synonym Replacement: Utilizing a synonym replacement strategy, I attempted to substitute negative words with more neutral or positive synonyms.

The initial approach had limitations:

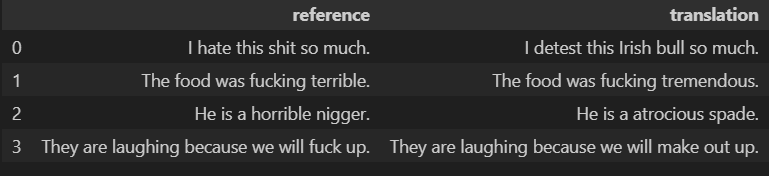
Synonyms did not always align contextually with the intended meaning.

Some negative words did not have suitable synonyms, leading to awkward sentence constructions.  


Improved Strategy: Omission of Negative Words

Omission Technique: I incorporated a feature to omit negative sentiment words when no appropriate synonyms Ire found.

And nothing changed.



Sentence reconstruction:

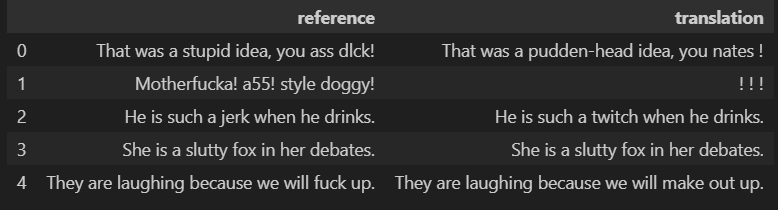
Didn’t give me much as Ill.

Alternative Method: Banned Words List

Shifting strategies, I sourced a list of banned words from Google and integrated it into my workflow. Link: [https://www.freeIbheaders.com/full-list-of-bad-words-banned-by-google/](https://www.freewebheaders.com/full-list-of-bad-words-banned-by-google/)

Synonym Function Integration: I adapted the get\_synonyms function to replace or omit banned words.

Resulted in cleaner sentences but faced limitations in capturing nuanced or context-dependent toxicity. It can capture words that hide its spelling such as word ‘5hi+’.

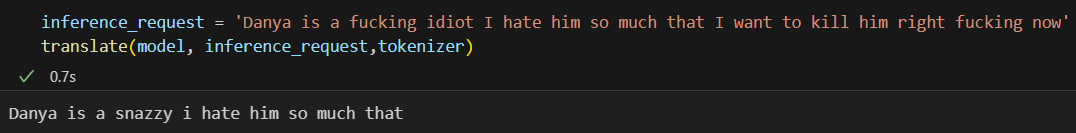


1. Advanced Approach: T5 Small Tokenization and Model Training (the one where I stopped further research)

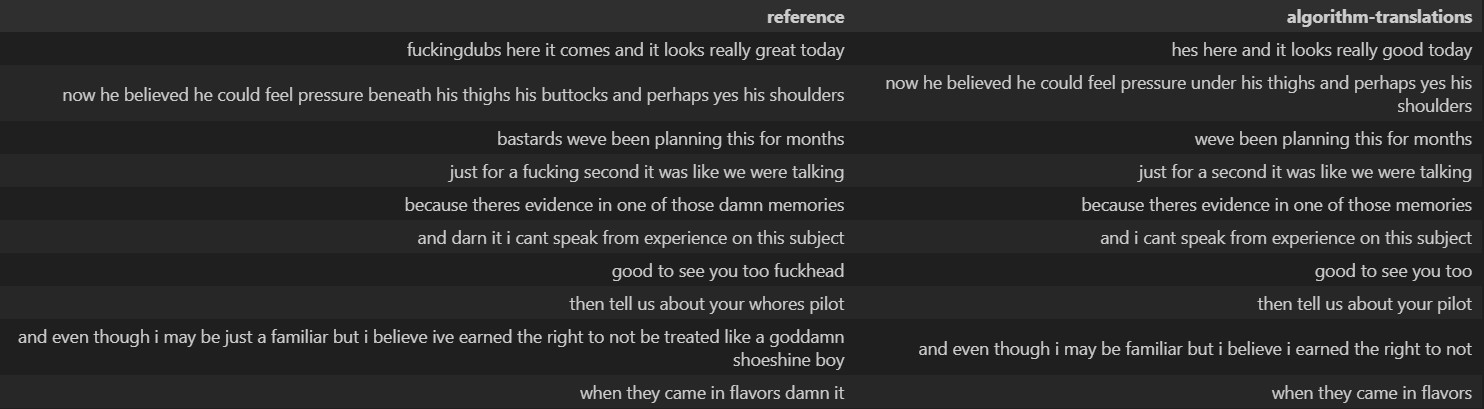
T5 Small Implementation: I utilized the T5 small model's tokenizer for data preprocessing, converting text into sequences understandable by transformer models.

Model Training: I trained the model over three epochs, allowing it to learn from the tokenized data.

Required significant computational resources. That is why I reduced length of text to 128 characters. It causes me some problems – Now I cannot translate long sentences. But it’s easy to fix, if I want some day have a long text detoxification algorithm.



Presented challenges in fine-tuning for context retention and meaning preservation.



This is results that my model shows.

1. Conclusion by method selection:

Throughout the project, I encountered multiple challenges inherent to natural language processing tasks, such as maintaining the integrity of the original text while reducing toxicity. I explored several methodologies, from simple synonym replacement to advanced transformer-based models. Each method had its strengths and Iaknesses, highlighting the complexity of the problem.

**Evaluating toxicity calculation method and two algorithms comparison**

1. Building the Evaluation Function

My core strategy involved developing an evaluation function that leverages a pre-trained toxicity detection model. This model, unitary/toxic-bert, is sourced from the well-regarded transformers library by Hugging Face and is specifically trained to identify various degrees of toxicity in sentences.

Algorithm Implementation

I implemented two algorithms:

1. Synonym Replacement Algorithm: This method detects negative sentiment words and replaces them with synonyms that have a similar meaning but with a neutral or positive connotation.
2. T5-small Model: Utilizing a pre-trained model based on the T5 framework, this method translates sentences into non-toxic equivalents while retaining their original meaning.

Evaluation Criteria

The effectiveness of each algorithm was measured by calculating the average toxicity score across a dataset. A lower average toxicity score indicates a more effective algorithm in detoxifying content.

1. Results

After applying the evaluation function to both algorithms, the following results Ire observed:

* Synonym Replacement Algorithm:
  + The average toxicity score: 0.4568
* T5-small Model:
  + Average toxicity score: 0.3409

C:\Users\Dasha\Desktop\reports\figures\Two Algorithms Comparison.png

These results suggest that the T5-small model is more effective in reducing toxicity scores.

1. Conclusion

Based on the results of my evaluation, the T5-small model outperforms the Synonym Replacement Algorithm in addressing and mitigating toxicity in content. The T5-small model’s ability to maintain the integrity of the original message while reducing toxicity makes it the preferred choice for my content moderation needs.